



Ultra Cloud Core Common Data Layer リリース 1.12 設定およびアドミニストレーション ガイド

最終更新：2025 年 7 月 17 日

シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先：シスコ コンタクトセンター

0120-092-255（フリーコール、携帯・PHS含む）

電話受付時間：平日 10:00～12:00、13:00～17:00

<http://www.cisco.com/jp/go/contactcenter/>

【注意】シスコ製品をご使用になる前に、安全上の注意（www.cisco.com/jp/go/safety_warning/）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS” WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2024-2025 Cisco Systems, Inc. All rights reserved.



目次

Full Cisco Trademarks with Software License ?

はじめに :

このマニュアルについて ix

使用する表記法 ix

第 1 章

Cisco Common Data Layer 1

概要 1

マニュアルの変更履歴 2

アーキテクチャ 2

CDL エンドポイント 5

単一のコンピューティングまたはノード障害 6

複数のコンピューティングまたはノード障害 6

スロット 6

単一のコンピューティングまたはノード障害 10

複数のコンピューティングまたはノード障害 10

インデックス作成 10

単一のコンピューティングまたはノード障害 12

複数のコンピューティング障害 12

ETCD 13

Kafka 13

Zookeeper 13

Mirror Maker 13

リモートサイトのモニタリング 13

機能説明 13

リモートサイト接続の設定 14

トラブルシューティング情報	14
CDL 導入モデル	14
コールフロー	16
プライマリキーによるレコードの検索	17
一意のキーによるレコードの検索	17
レコードの作成	18
レコードの更新	19
レコードの削除	20
タイマーが切れた時の NF への通知	21
Geo レプリケーション：作成	22
Geo レプリケーション：更新	22
Geo レプリケーション：削除	23
古いインデックスレコードの識別	24
ネットワーク機能（NF）を介した CDL の展開	26
CDL Geo レプリケーション（GR）の展開	26
CDL GR の前提条件	26
Geo レプリケーション（GR）用の CDL の展開	30
Geo レプリケーション（GR）のフェールオーバーの通知	34
ピアサイトの GR フェールオーバー通知レコード	35
リモートインデックス同期のトリガー	36
機能の概要と変更履歴	38
機能説明	39
IPv6 サポートの設定	39
CDL アラーム	43
CDL ゾーンのアップグレード	44
機能説明	44
トポロジの分散制約	45
制限事項と制約事項	46
CDL のオーバーロードの防止	46
CDL の最大キャパシティ	47
CDL でのオーバーロードのアラートのトリガー	47

オーバーロードの防止の設定	48
アラートの確認	50
CDL のラック変換	51
機能説明	51
CDL 変換の MOP	51
トラブルシューティング	59
CDL 用の FindAll および FindAllNotify クエリの機能拡張	60
機能説明	60
機能の仕組み	60
ネットワーク ポリシー設定	63
機能の概要と変更履歴	63
要約データ	63
マニュアルの変更履歴	63
機能説明	63
CDL ネットワークポリシーの設定	63
トラブルシューティング情報	64
CDL インデックスが正確にレプリケートされない	64
CDL の操作が失敗しているのに、接続は成功する	65
CDL のポッドがダウンしている	66
準備中の状態の Mirror Maker ポッド	67
CDL からのレコード消去の通知が早い、または遅延する	67
CDL でセッションが蓄積する、またはオーバーロードする	68
ユーザー定義のスケジュール設定とポッドの配置	68
モニタリング	71
Grafana ダッシュボードを介した CDL のモニタリング	71
CDL のステータスとレコードの確認	74

第 2 章

CDL のバックアップと復元 77

CDL Ops-Center の設定へのアクセス 77

バックアップと復元 78

第 3 章

CDL のメトリック 81

概要 81

CDL カテゴリ 82

第 1 部 :

CDL 設定モードのコマンドリファレンス 125

第 4 章

CDL 設定モードのコマンドリファレンス 127

cdl 129

cdl actions backup 130

cdl actions restore 131

cdl api 131

cdl datastore 132

cdl datastore actions remote-index-sync start 133

cdl datastore actions remote-index-sync start map-id 134

cdl datastore actions remote-index-sync start slice-name 134

cdl datastore actions remote-index-sync status 134

cdl datastore clear sessions 135

cdl datastore clear sessions filter 135

cdl datastore clear sessions in-filter 136

cdl datastore endpoint 137

cdl datastore endpoint settings 139

cdl datastore features index-overwrite-detection unique-keys-prefix 139

cdl datastore features instance-aware-notification 140

cdl datastore features instance-aware-notification system-id 140

cdl datastore features overload-protection 141

cdl datastore features slot-stale-session-detection 143

cdl datastore index 144

cdl datastore kafka 145

cdl datastore kafka describe 147

cdl datastore kafka describe consumer-groups 148

cdl datastore kafka describe topics 148

cdl datastore kafka external-ip 148

cdl datastore kafka label-config	149
cdl datastore kafka list	149
cdl datastore kafka list consumer-groups	150
cdl datastore kafka list topics	150
cdl datastore kafka log-compaction	150
cdl datastore kafka reset-offsets	151
cdl datastore kafka reset-offsets consumer-group	151
cdl datastore kafka ssl-settings	152
cdl datastore metrics	152
cdl datastore mirror-maker	153
cdl datastore overload-protection	153
cdl datastore reset-offsets-dry-run	154
cdl datastore reset-offsets-dry-run consumer-group	154
cdl datastore session features remote-site-connection-monitoring	155
cdl datastore show indexes	155
cdl datastore show sessions count detailed	156
cdl datastore show sessions count detailed filter	157
cdl datastore show sessions count detailed in-filter	158
cdl datastore show sessions count summary	158
cdl datastore show sessions count summary filter	159
cdl datastore show sessions count summary in-filter	160
cdl datastore show sessions detailed	161
cdl datastore show sessions detailed filter	162
cdl datastore show sessions detailed in-filter	163
cdl datastore show sessions summary	163
cdl datastore show sessions summary filter	165
cdl datastore show sessions summary in-filter	165
cdl datastore show status	166
cdl datastore slot	167
cdl datastore slot notification	168
cdl datastore slot notification metrics	170
cdl datastore zookeeper	171
cdl label-config	172
cdl label-config endpoint	173

cdl label-config index map	173
cdl label-config slot map	174
cdl logging	174
cdl logging logger	175
cdl remote-site	176
cdl remote-site db-endpoint	176
cdl remote-site kafka-server	177
cdl ssl-config	177
cdl ssl-config certs	178



このマニュアルについて



- (注) この製品のマニュアルセットは、偏向のない言語を使用するように配慮されています。このドキュメントセットでの偏向のない言語とは、年齢、障害、性別、人種的アイデンティティ、民族的アイデンティティ、性的指向、社会経済的地位、およびインターセクショナリティに基づく差別を意味しない言語として定義されています。既存の偏向のある用語は置換していますが、製品ソフトウェアのユーザーインターフェイスにハードコードされている言語、RFPのドキュメントに基づいて使用されている言語、または参照されているサードパーティ製品で使用されている言語によって、ドキュメントに例外が存在する場合があります。

ここでは *Common Data Layer* について、その構成とマニュアルの表記法を説明します。

このガイドでは Cisco Common Data Layer (CDL) について説明しており、これにはインフラストラクチャとインターフェイス、機能の説明、仕様の準拠、セッションフロー、設定手順、およびシステムのモニタリングと障害対応のための CLI コマンドが含まれています。

- [使用する表記法](#) (ix ページ)

使用する表記法

次の表に、このマニュアル全体で使用される表記法を示します。

通知タイプ	説明
情報メモ	重要な機能または手順に関する情報を提供します。
注意	プログラム、デバイス、またはシステムに損傷を与えるおそれがあることを注意喚起します。

通知タイプ	説明
警告	人身傷害または死亡事故のおそれがあることを警告します。また、電氣的障害のおそれがあることを警告する場合もあります。

書体の表記法	説明
スクリーンディスプレイとして表されるテキスト	この書体は、端末画面に表示されるディスプレイを表します。次に例を示します。 ログイン：
commands として表現されるテキスト	この書体は、入力したコマンドを表します。次に例を示します。 show ip access-list このマニュアルでは、コマンドの完全表記に常に小文字を使用しています。コマンドには、大文字と小文字の区別はありません。
command variable として表されるテキスト	この書体は、コマンドの一部である変数を表します。次に例を示します。 show card slot_number <i>slot_number</i> は、目的のシャーシのスロット番号を表す変数です。
メニュー名またはサブメニュー名として表されるテキスト	この書体は、ソフトウェアアプリケーション内でアクセスするメニューとサブメニューを表します。次に例を示します。 [File] メニュー、[New] の順にクリックしてください。



第 1 章

Cisco Common Data Layer

- [概要 \(1 ページ\)](#)
- [アーキテクチャ \(2 ページ\)](#)
- [リモートサイトのモニタリング \(13 ページ\)](#)
- [CDL 導入モデル \(14 ページ\)](#)
- [CDL ゾーンのアップグレード \(44 ページ\)](#)
- [CDL のオーバーロードの防止 \(46 ページ\)](#)
- [CDL のラック変換 \(51 ページ\)](#)
- [CDL 用の FindAll および FindAllNotify クエリの機能拡張 \(60 ページ\)](#)
- [ネットワーク ポリシー設定 \(63 ページ\)](#)
- [トラブルシューティング情報 \(64 ページ\)](#)
- [モニタリング \(71 ページ\)](#)

概要

Cisco Common Data Layer (CDL) は、クラウドネイティブなすべてのアプリケーションに対応する、高性能の次世代キーバリュー (KV) データストアレイヤです。これらのアプリケーションは、高可用性 (HA) と Geo HA 機能を備えた状態管理として CDL を使用します。CDL は以下を提供します。

- AMF、SMF、PCF などのさまざまなネットワーク機能 (NF) にまたがる共通のデータストアレイヤ：マイクロサービス。
- 低遅延の読み取りと書き込みを実現するマルチプライマリサポート。
- 純粋なインメモリストレージ。
- セッション関連のタイマーを実行し、タイマーが切れた時に NF に通知します。
- 高速フェールオーバーによる高可用性と Geo 冗長性

マニュアルの変更履歴

表 1: マニュアルの変更履歴

改訂の詳細	リリース
• CDL は古いインデックスレコードについて、識別と必要なアクションの実行をサポートします。 • ピアサイトが隔離サイトの GR フェールオーバー通知を処理できるようになっています。 • CDL は、インデックスをリモートピアと同期させるためのユーティリティを提供します。 • Grafana ダッシュボードの拡張により、GR 接続ステータスが表示されます。	2020.02.3

アーキテクチャ

CDL は次のモードで展開できます。

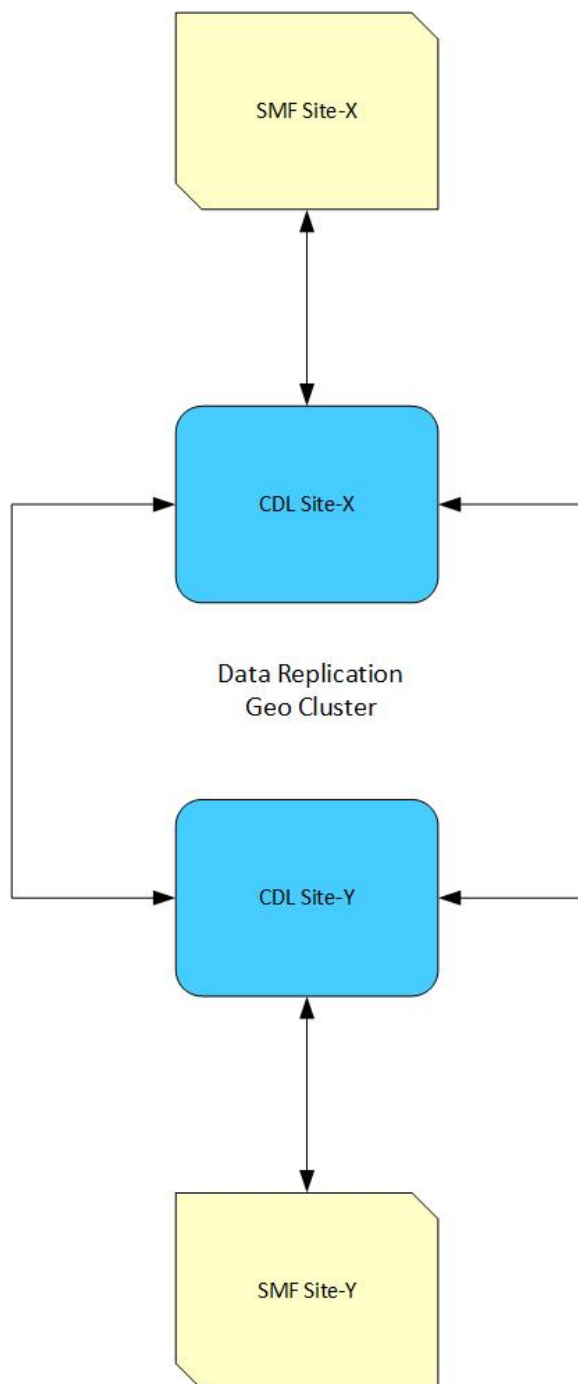
- HA のみ
- Geo HA



(注) HA のみのデプロイメントでは、ローカルサイトでブレード障害に対する冗長性が提供されません。ただし、完全なサイトまたは K8s クラスタに到達できない場合、HA のみのデプロイメントではセッション冗長性は提供されません。

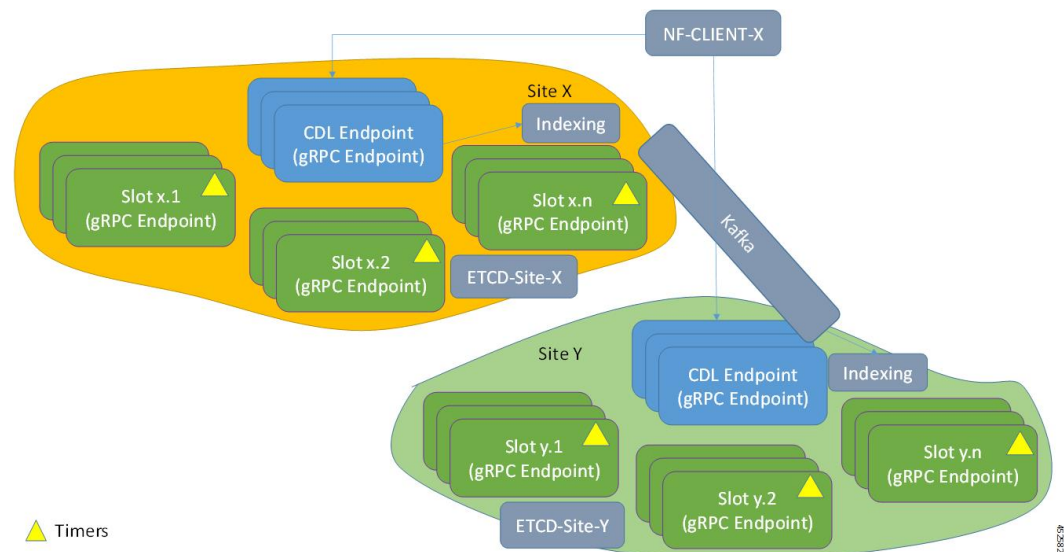
次の図は、CDL 導入モデルのハイレベルアーキテクチャを示しています。

図 1: CDL 導入モデル



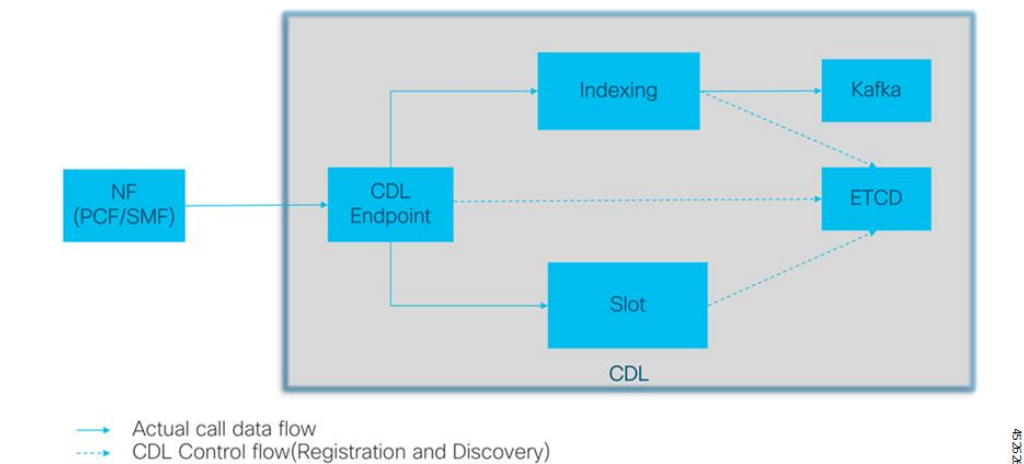
45281

図 2: CDL マイクロサービス



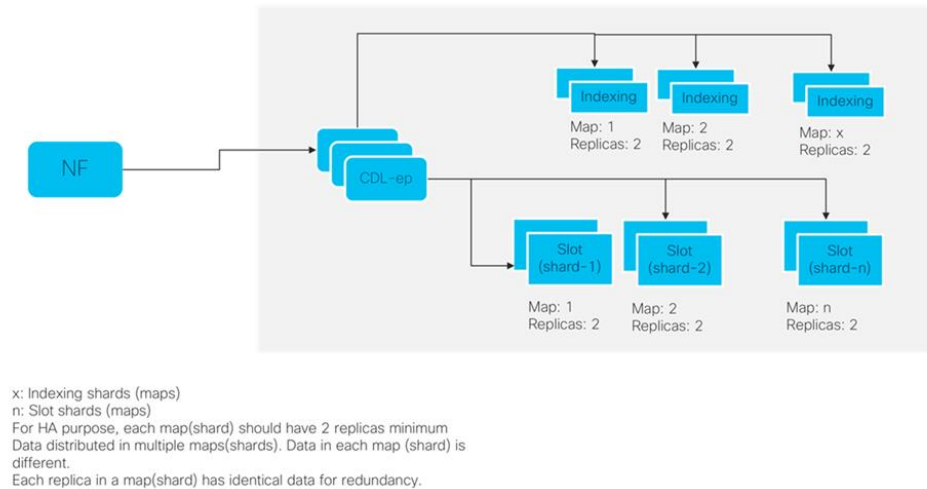
次の図は、CDL のハイレベルアーキテクチャを示しています。

図 3: CDL のアーキテクチャ



次の図は、主要なコンポーネントを含む CDL のハイレベルアーキテクチャを示しています。

図 4: 主要なコンポーネントを含む CDL アーキテクチャ



CDL エンドポイント

CDL エンドポイントポッドは、NF アプリケーションから CDL に対する要求を受信するためのフロントエンドです。CDL エンドポイントは、データベースサービス要求を処理するために、NF クライアントに向けて HTTP2 インターフェイスを介して gRPC を公開します。各ポッドは、次の属性で始まります。

- **systemID** : このパラメータは、サイト ID (例 : Site) を指定します。
- **clusterID** : このパラメータは、サイト内の一意のデータストア ID (例 : X、Y) を指定します。

CDL エンドポイントは、NF アプリケーションから作成、読み取り、更新、削除 (CRUD) 要求を受信します。これらの操作を実行するために、GRPC を介して CDL インデックスおよびスロットポッドと通信します。

CDL エンドポイントは、サイト内の他のポッドについて、*etcd* を使用して自動的に学習します。また、Cisco Data Store K8s ポッドは本質的にステートレスであり、セッションのステイキ性は維持されません。CDL エンドポイントは NF アプリケーションから要求を受信し、GRPC を介して内部的にスロットポッドおよびインデックスポッドと通信して、それに応じて要求を処理します。処理が完了すると応答を返します。

CDL エンドポイントが作成、削除、または更新セッションを受信すると、受信コンテナはそれをスロットポッドおよびインデックスポッドにレプリケートし、「n」個の書き込みがインデックスおよびスロットのマイクロサービスによって確認 (ACKed) された場合にのみ、応答を送信します。



(注) 実稼働環境で CDL エンドポイントポッドを展開するには、最大 4 つの仮想 CPU (vCPU) が必要です。

単一のコンピューティングまたはノード障害

単一のコンピューティングまたはノードで障害が発生すると、**cdl-ep** ポッドの 1 つがダウンし、他のセッションノードでスケジュールを試行します。他のすべてのセッションノードに **cdl-ep** がすでにある場合、**cdl-ep** ポッドにノードのアンチアフィニティが定義されているため、ポッドのスケジュール設定は行われません。したがって、単一のノード障害によって残りの **cdl-ep** ポッドがトラフィックを処理することになり、多くの要求による負荷がかかる可能性があります。ノード障害から回復した後、以前は保留状態にあった **cdl-ep** ポッドが、回復したノードで要求を処理するためのスケジュール設定を開始します。

複数のコンピューティングまたはノード障害

すべてのセッションノードに障害が発生すると、それに合わせてすべての **cdl-ep** ポッドがダウンします。Geo 高可用性 (GeoHA) が設定されている場合、NF アプリケーションはリモートサイトと通信し、ローカルサイトでセッションの処理を続行します。ただし、高可用性 (HA) シナリオでは、これにより完全なダウンタイムが発生します。

スロット

CDL スロットポッドには、実際のセッションデータが保存されます。CDL エンドポイントはクラスタ内のすべてのスロットポッドに接続し、セッションデータをすべてのポッドにレプリケートします。これらのマイクロサービスは、内部 gRPC インターフェイスを Cisco Data Store に公開するために展開された K8s ポッドです。各ポッドは、次の属性で始まります。

- **systemID** : このパラメータは、サイト ID (例 : Site-1) を指定します。
- **clusterID** : このパラメータは、サイト内の一意のデータストア ID (例 : Session) を指定します。
- **mapID** : このパラメータは、クラスタ内のレプリカセット ID を指定します。(例 : map-1、map-2、..map-n)。
- **instanceID** : このパラメータは、レプリカセット内のインスタンス ID を指定します。(例 : map-1.instance-1、map-1.instance-2)

各スロットポッドは、限られた数のセッションと、保存されているセッションデータの事前に割り当てられたメモリを保持します。また、レプリカセット (**mapID**) 内の各レプリカには、定義されたアンチアフィニティルールがあり、同じブレードフォームが同じレプリカセットの複数のメンバーまたはインスタンスをホストするのを防ぎます (ブレードまたはノードに障害が発生した場合の高可用性のため)。各スロットポッドは、タイマーと最後に更新された TS を維持します。スロットポッドは、競合が検出された場合、タイマーが切れたときにアクションを実行するために、クライアント NF への通知コールバックを生成します。



(注) 実稼働環境でスロットポッドを展開するには、最大で2つのvCPUが必要です。

ポッドのフェールオーバーと回復が発生した場合、スロットポッドは以下から回復します。

- **ローカルレプリカメンバー**：スロットはgRPCストリームから一括で直接読み取り、データを回復します。
- **リモートレプリカメンバー**：同期に使用できるローカルレプリカがない場合、スロットは同じマップのリモートサイトインスタンスからデータを読み取ります。

次の図は、ローカルピアおよびリモートピアからのスロットリカバリプロセスを示しています。

図 5: ローカルピアからのスロットリカバリ

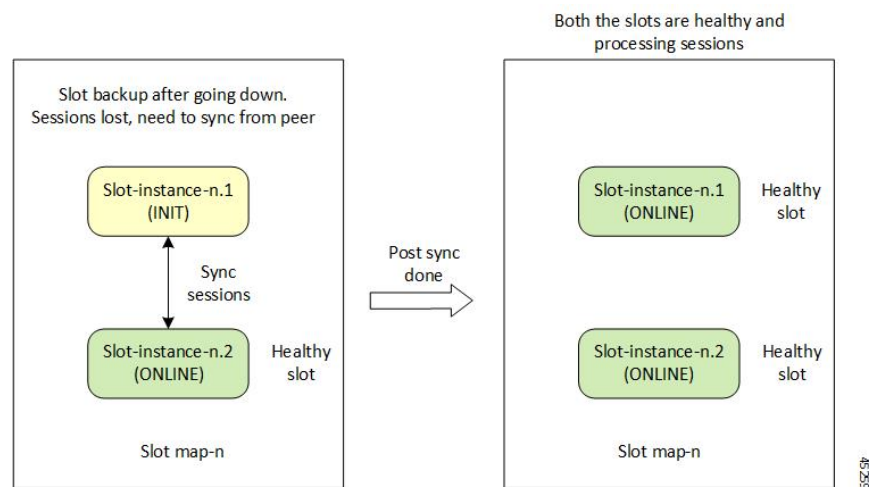
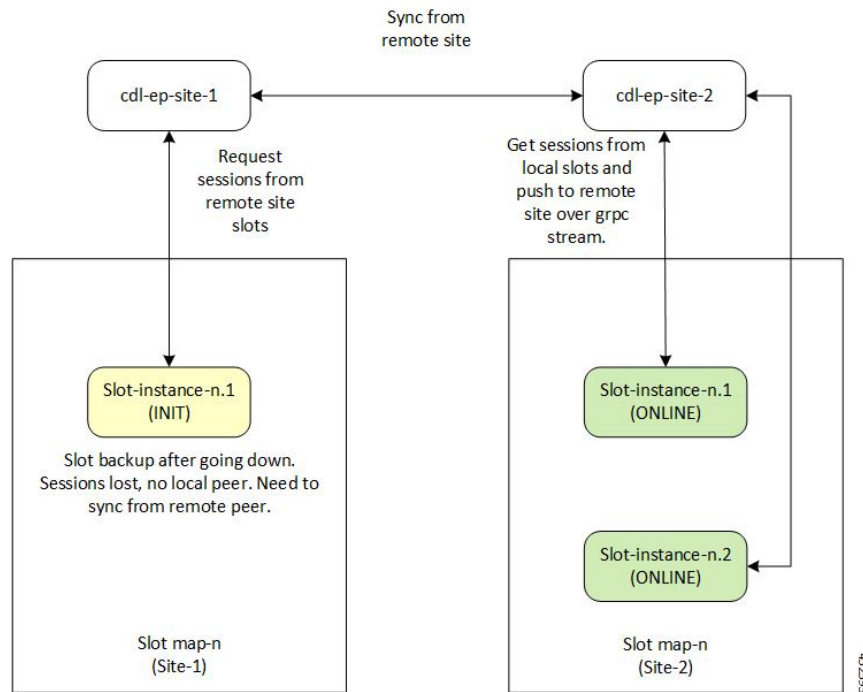


図 6: リモートピアからのスロットリカバリ



データのスライシング

データのスライシングでは、CDL がスライスとして論理的に分割され、ネットワーク機能 (NF) から受信したスライス名に基づいてセッションデータが保存されます。

データのスライシングを使用すると、1 つ以上の NF がさまざまなタイプのセッションデータを CDL の専用スライスに保存できます。

スライス名が設定されていない場合は、**session** というデフォルトのスライス名が使用されます。

設定は次のとおりです。

```
cdl datastore <datastore name> slice-names [ <sliceName 1> <sliceName 2> ... <sliceName n> ]
```

サンプル設定を次に示します。

```
cdl datastore session slice-names [ session1 session2 ]
```



- (注)
- スライス名が NF の **ops-center** または CDL の **ops-center** で設定されている場合、NF からのすべての要求に有効なスライス名が必要です。スライス名が設定されているものと違うか空の場合、エラーコードが表示され要求は拒否されます。
 - スライス名が設定されていない場合、NF 要求はデフォルトの **session** にルーティングされます。
 - デプロイメント後の実行中のシステムでは、スライス名を更新できません。

CDL スロットデータの削除

特定のシナリオでは、CDL レコードがスロットでは見つかりませんが、インデックスポッドでは見つかりません。そのようなレコードに関するスロットからアプリケーションへの通知では、値が正しく受信されません。インデックスデータが削除されない場合、スロット内のレコードは削除されません。

CDL スロットデータを削除する前に、次のことを確認してください。

- アプリケーションへの通知数がしきい値（デフォルト値の 3）を超えた場合、レコードが古い状態であると考えられます。
- これにより、いずれかのインデックスポッド（ローカルまたは Geo リモートサイト）で対応するレコードを見つけるために、検証チェックがトリガーされます。
- インデックスのマップ ID に不一致がある場合、またはマップ ID がすべてのインデックスポッドで見つからない場合、クリーンアップが呼び出され、ローカルサイトとリモートサイトのレコードが削除されます。

古いレコードを削除するために、次のパラメータが導入されました。

disable-auto-deletion : true に設定すると、古い CDL レコードは削除されません。古いレコードの自動削除は、デフォルトで有効になっています。

notification-retry-count : アプリケーションから更新を受信せずにアプリケーションに送信するタイマー切れ通知の最小再試行回数を指定します。**notification-retry-count** の回数を超えても更新を受信しない場合、CDL はスロットレコードが古いかどうかのチェックに進みます。デフォルトの回数は 3 です。

CDL のサンプル設定は次のとおりです。

古いスロットレコードの自動削除機能を無効にするには、次の手順を実行します。

```
cdl datastore session
features slot-stale-session-detection disable-auto-deletion true
exit
```

notification-retry-count は、5 などの新しい値に変更できます。これは、タイマー切れ通知を 5 回試行した後、データが古いかどうかのチェックに進むことを示しています。

```
cdl datastore session
features slot-stale-session-detection notification-retry-count 5
exit
```

トラブルシューティング

エンドポイントおよびスロットポッドで古い CDL スロットデータの障害対応ログを有効にするには、次の設定を使用します。

```
cdl logging logger ep.staleRecord.session
level info
exit

cdl logging logger slot.staleRecord.session
level info
exit
```

単一のコンピューティングまたはノード障害

単一のコンピューティング障害が発生した場合、CDL は一部のスロットおよびインデックスポッドの 1 つ少ない *cdl-ep* ポッドとレプリカで機能し続けます。単一のレプリカにより、CDL スロットマップはすべての要求の処理を続行します。サービスを停止したノードがサービスに戻ると、「[ローカルピアからのスロットリカバリ](#)」の図に示されているように、ノードのスロットポッドがそれぞれのピアと同期します。

複数のコンピューティングまたはノード障害

複数のコンピューティング障害が発生した場合、スロットマップのすべてのレプリカがダウンします。たとえば、2 つのノードがダウンしている場合、スロットマップの両方のレプリカがダウンします（スロットレプリカ数のデフォルト値は 2）。これにより、ローカルのレプリカセット全体が機能不全となり、すべての *cdl-ep* ポッドのシャットダウンがトリガーされます。また、GeoHA が設定されている場合は、CDL Geo レプリケーション（GR）がトリガーされます。GeoHA が設定されている場合、NF アプリケーションはリモートサイトの *cdl-ep* と通信して要求を処理します。GeoHA 設定がない場合、最終的にサービスのダウンタイムが発生します。

障害が発生したノードが回復すると、スロットポッドとスロットレプリカセットが回復し、リモートサイトから同期されます（障害発生中に両方のレプリカがダウンした場合）。これは、GeoHA が設定されている場合にのみ発生します。[リモートピアからのスロットリカバリ](#)の図は、ローカルのレプリカセット全体がダウンした場合のリモートピアからの最初の同期を示しています。

インデックス作成

インデックス作成コンテナには、インデックス作成データが含まれます。インデックスポッドには、次の 2 つの重要な情報が保存されます。

- スロットマップ ID のプライマリーキー。
- セカンダリキーからプライマリーキーへの固有のマッピング。

インデックスエントリは、セッション接続時に作成され（使用例の場合）、切断時に削除されます。インデックス作成では、インメモリ KV ストアを使用してキーをメモリに保存し、マルチプライマリ書き込みを提供します。インデックス作成書き込みログ（設定操作と削除操作）も、リモートサイトのレプリケーション用に **Kafka** ポッドに書き込まれます。インデックスポッドの 1 つが **Kafka** でログを書き込み、リモートサイトでそのログを受信します。対応する操作もリモートサイトで実行されます。



（注） 実稼働環境でインデックスポッドを展開するには、最大で 2 つの vCPU が必要です。

ポッドのフェールオーバーと回復が発生した場合、インデックスポッドは以下から回復します。

- **ローカルレプリカメンバー**：インデックスは gRPC ストリームから一括で直接読み取り、データを回復します。
- **リモートレプリカメンバー**：同期に使用できるローカルレプリカがない場合、インデックスは同じマップのリモートサイトインスタンスからデータを読み取ります。

次の図は、ローカルピアおよびリモートピアからのインデックス リカバリ プロセスを示しています。

図 7: ローカルピアからのインデックスリカバリ

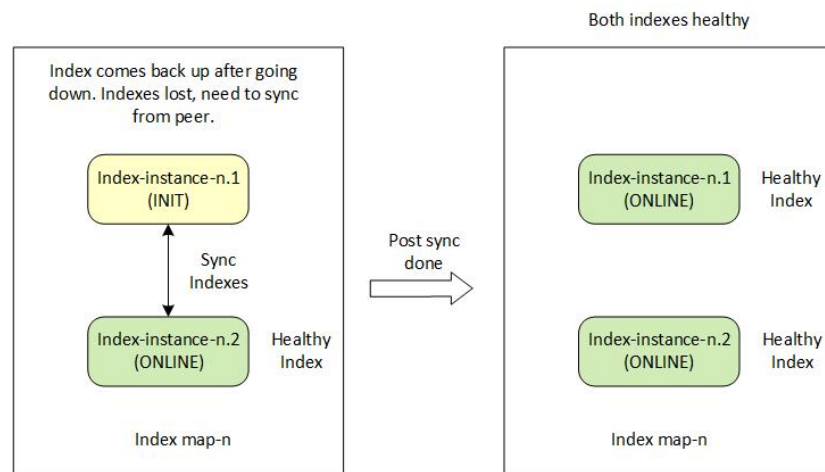
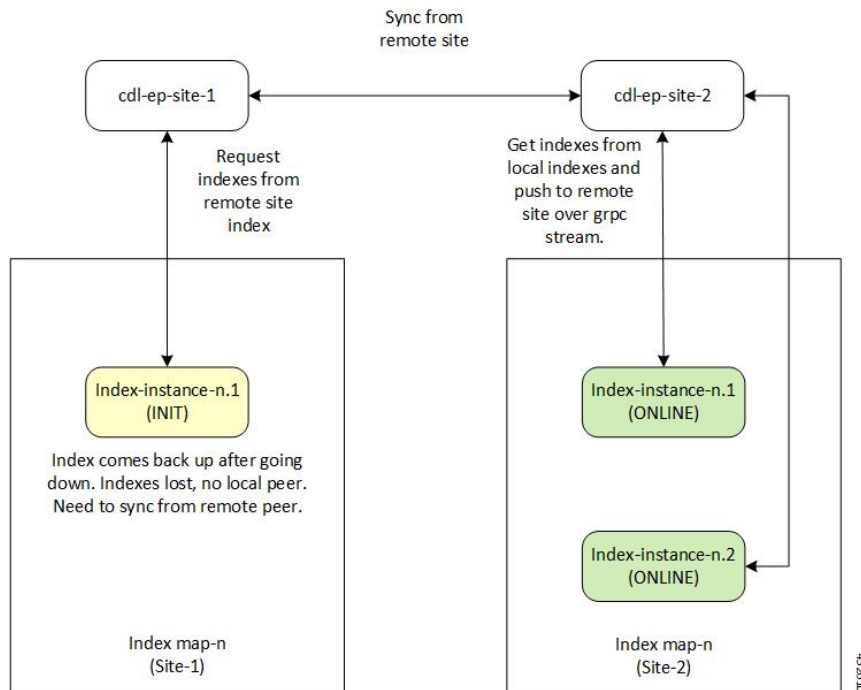


図 8: リモートピアからのインデックスリカバリ



単一のコンピューティングまたはノード障害

単一のコンピューティング障害が発生した場合、CDL は一部のスロットおよびインデックスポッドの 1 つ少ない `cdl-ep` ポッドとレプリカで機能し続けます。単一のレプリカにより、CDL スロットマップはすべての要求の処理を続行します。サービスを停止したノードがサービスに戻ると、「ローカルピアからのインデックスリカバリ」の図に示されているように、ノードのインデックスポッドがそれぞれのピアと同期します。

複数のコンピューティング障害

複数のコンピューティング障害が発生した場合、スロットマップのすべてのレプリカがダウンします。たとえば、2 つのノードがダウンしている場合、スロットマップの両方のレプリカがダウンします（スロットレプリカ数のデフォルト値は 2）。これにより、ローカルのレプリカセット全体が機能不全となり、すべての `cdl-ep` ポッドのシャットダウンがトリガーされます。また、GeoHA が設定されている場合は、CDL Geo レプリケーション（GR）がトリガーされます。GeoHA が設定されている場合、NF アプリケーションはリモートサイトの `cdl-ep` と通信して要求を処理します。GeoHA 設定がない場合、最終的にサービスのダウンタイムが発生します。

障害が発生したノードが回復すると、インデックスポッドとインデックスレプリカセットが回復し、リモートサイトから同期されます（障害発生中に両方のレプリカがダウンした場合）。これは、GeoHA が設定されている場合にのみ発生します。リモートピアからのインデックスリカバリの図は、ローカルのレプリカセット全体がダウンした場合のリモートピアからの最初の同期を示しています。

ETCD

CDL は、DB サービス検出として *etcd*（オープンソースのキーバリューストア）を使用します。CDL ポッド（エンドポイント、スロット、またはインデックス）が開始、強制終了、またはシャットダウンされると、状態のパブリッシュによってイベントが *etcd* に更新されます。

CDL エンドポイントは、スロットポッドまたはインデックスポッドがいつ起動または停止したかを知る必要があります。したがって、スロットポッドおよびインデックスポッドのイベントの通知に *etcd* で登録されます。

同様に、CDL スロットポッドまたはインデックスポッドは、そのピアレプリカポッド（同じ *map-id* を持つ）がいつ起動または停止したかを知る必要があります。したがって、対応するスロットまたはインデックス（同じ *map-id* を持つ）のポッドイベントの通知に *etcd* で登録されます。

これにより、通知はこれらのイベントに登録された各ポッドに送信されます。また、キーイベントが追加または削除されると、ローカルマップが更新されます。*etcd* キャッシュは、ローカルサイトのイベントにのみ適用されます。

Kafka

Kafka ポッドは、インデックス作成のためにローカルレプリカ間とサイト間でデータをレプリケートします。サイト間のレプリケーションの場合、Kafak は MirrorMaker を使用します。Kafka ポッドは、高可用性のためにレプリカ数が少なくとも 2 に設定されたセッション *vms* に展開されます。

Zookeeper

Kafka は Zookeeper ポッドを使用して Kafka クラスタを管理し、Kafka ブローカとの調整を行います。

Mirror Maker

Mirror Maker ポッドは、インデックスデータをリモート CDL サイトに Geo レプリケートします。リモートサイトからデータを取得し、ローカルの Kafka サイトにパブリッシュして、適切なインデックス作成インスタンスを取得します。

リモートサイトのモニタリング

機能説明

CDL エンドポイントは、CDL Ping RPC を使用して 30 秒ごとにリモートサイトの接続（レプリケーション接続と内部運用接続）をモニターします。いずれかの接続に対して ping が 3 回失敗した場合は、その接続を再作成して古い接続を閉じます。

リモートサイトのモニタリングは設定可能で、デフォルトでは有効になっています。**cdl datastore session features remote-site-connection-monitoring enable** CLI コマンドを使用して、リモートサイトの接続を有効または無効にします。

リモートサイト接続の設定

リモートサイトの接続を有効または無効にするには、次の CLI コマンドを使用します。

```
config
  cdl datastore session features remote-site-connection-monitoring
  enable [ true | false ]
exit
```

トラブルシューティング情報

エンドポイントのリモートサイト接続のログを表示するには、次の設定を使用します。

```
cdl logging logger ep.remoteConnection.session
level trace
exit
```

CDL 導入モデル

このセクションでは、次のようなさまざまな CDL 導入モデルについて説明します。

- CDL HA
- Geo HA デプロイメント

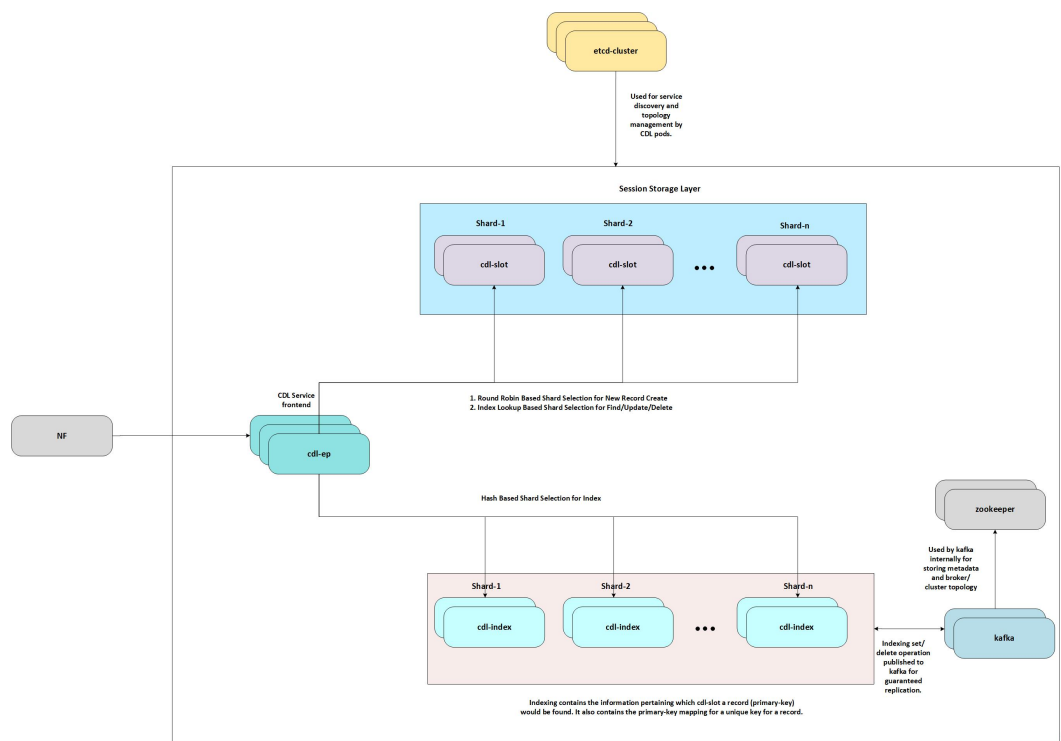
CDL HA デプロイメントには、次のものが含まれます。

1. 各 CDL ポッドは、高可用性を実現するために少なくとも 2 つのレプリカで構成されます。
2. CDL エンドポイントポッドは、サービス検出のために *etcd* を使用してローカルスロットとインデックスを検出します。
3. インデックスおよびスロットポッドのデータは、書き込みの拡張性のためにマップ（またはシャード）をまたいで分散され、各マップは高可用性のために少なくとも 1 つのレプリカを備えます。スロットとインデックスのレプリカのデフォルトかつ推奨のレプリカ数は 2 です。
4. 要求を受信した CDL エンドポイントは、データを書き込むまたは読み取るスロットシャードを選択します。新しいレコードを作成する場合は、ラウンドロビン方式でスロットマップを選択します。既存のセッションの場合、スロットマップはインデックス作成でプライマリキーを検索した後に選択されます。
5. セッションで新しいキー（プライマリまたは一意）が追加または削除された場合、CDL エンドポイントはインデックス作成ポッドにキーを送信します。インデックスマップの選択は、キーをハッシュし、インデックス内のキーを更新するために適切なインデックスマップ

プに送信することによって行われます。プライマリキーの場合、インデックス作成ポッドはセッションが保存されているスロット **map-id** を保存します。一意のキーの場合、インデックス作成ポッドはセッションが保存されているプライマリキーを保存します。

6. CDLエンドポイントは、スロットメモリのデータを書き込むまたは読み取るために、選択されたスロットに作成、更新、削除、または検索要求を転送します。要求が正常に処理されると、スロットはエンドポイントに対して適切な応答を生成します。
7. 各インデックス作成シャードには、ローカルおよびリモートのレプリケーションを保証するために、インデックス作成操作を **Kafka** バスにパブリッシュするリーダーが含まれています。
8. 各インデックス作成ポッドインスタンスは、インデックス作成イベントの **Kafka** バスをリッスンします。設定または削除イベントを受信すると、イベントのタイムスタンプをすでに存在するインデックスのタイムスタンプと比較します。**Kafka** イベントのタイムスタンプが現在のインデックスのタイムスタンプよりも大きい場合、インデックス作成ポッドに操作が適用されます。それ以外の場合は、**Kafka** からのイベントは無視されます。

図 9: CDL HA のデプロイメント



CDL Geo HA デプロイメントには、次のものが含まれます。

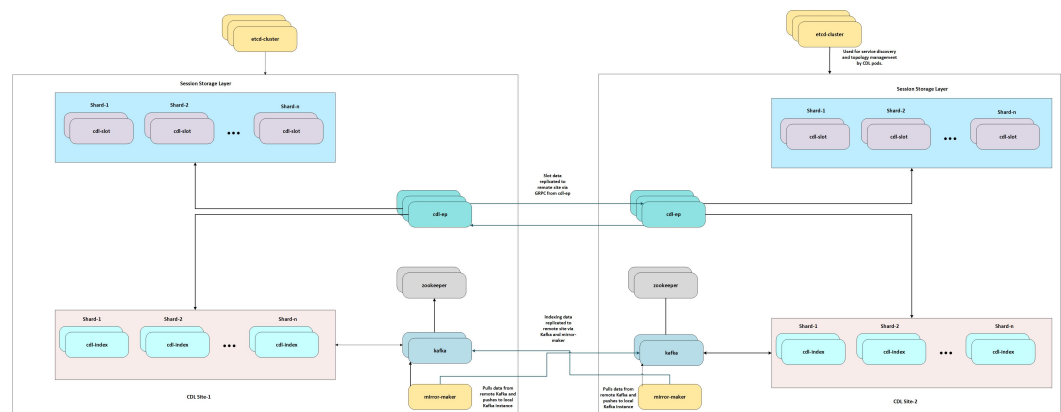
1. CDL Geo HA デプロイメントの場合、各サイトは Geo 冗長性を実現するためにリモートサイトで設定されます。

2. CDLエンドポイントは、ローカルのスロットとインデックスにデータを送信するだけでなく、GRPCを介してリモートサイトの *cdl-endpoint* にも要求を送信し、リモートサイトのスロットデータをレプリケートします。
3. リモートサイトで作成、更新、または削除要求を受信すると、CDLエンドポイントは、元のサイトで選択されていたものと同じ *map-id* で要求をスロットに転送します。
4. インデックス作成データのレプリケーションは、Mirror-makerによって実現されます。リモートサイトのMirror-makerは、他のサイトのKafkaからのデータを消費し、そのデータをローカルのインデックスポッドにレプリケートするためにローカルのKafkaバスにプッシュします。



(注) Geo HA デプロイメントの場合、2つのGeoサイト間の推奨ラウンドトリップ時間 (RTT) は50ミリ秒です。

図 10: CDL Geo HA のデプロイメント



コールフロー

このセクションでは、次のコールフローについて説明します。

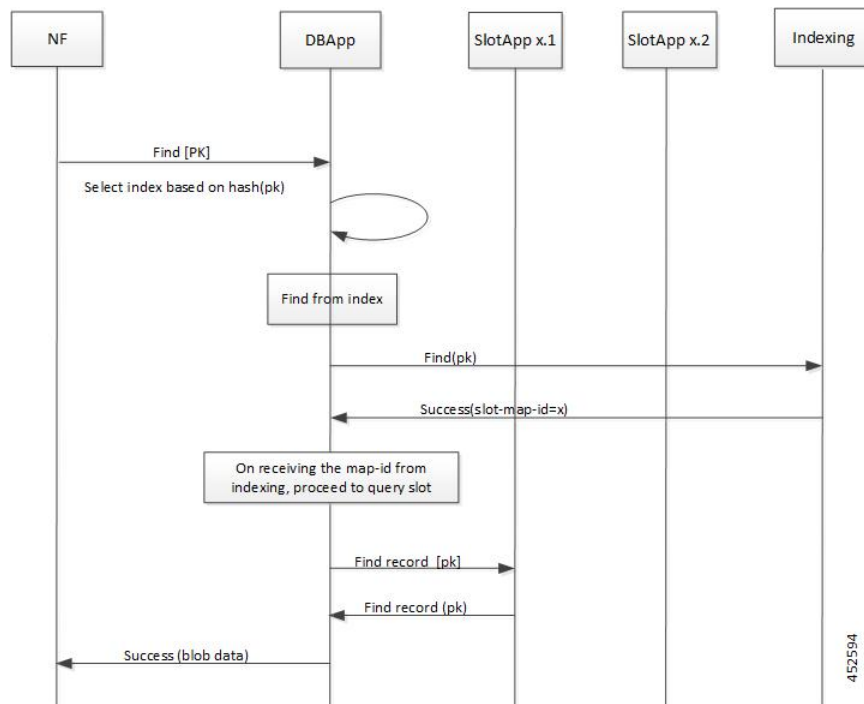
- プライマリキーによるレコードの検索
- 一意のキーによるレコードの検索
- レコードの作成
- レコードの更新
- レコードの削除
- タイマーが切れた時の NF への通知
- Geo レプリケーション：作成 (GEO)

- Geo レプリケーション：更新（GEO）
- Geo レプリケーション：削除（GEO）

プライマリキーによるレコードの検索

このセクションでは、プライマリキーでレコードを検索するコールフローについて説明します。

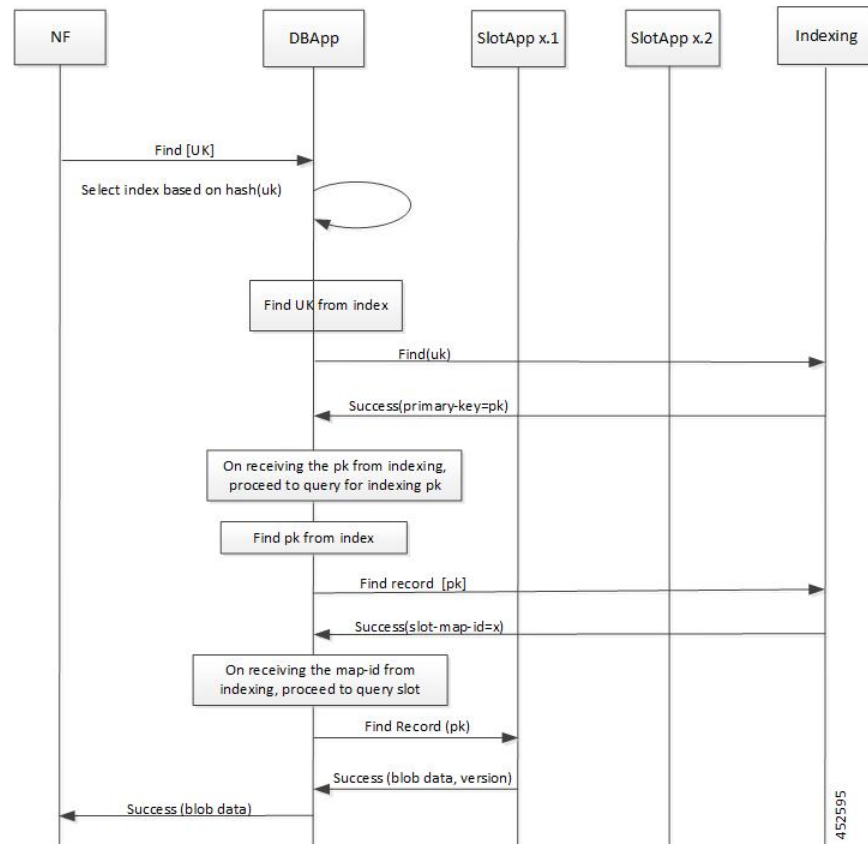
図 11: プライマリキーによるレコードの検索：コールフロー



一意のキーによるレコードの検索

このセクションでは、一意のキーでレコードを検索するコールフローについて説明します。

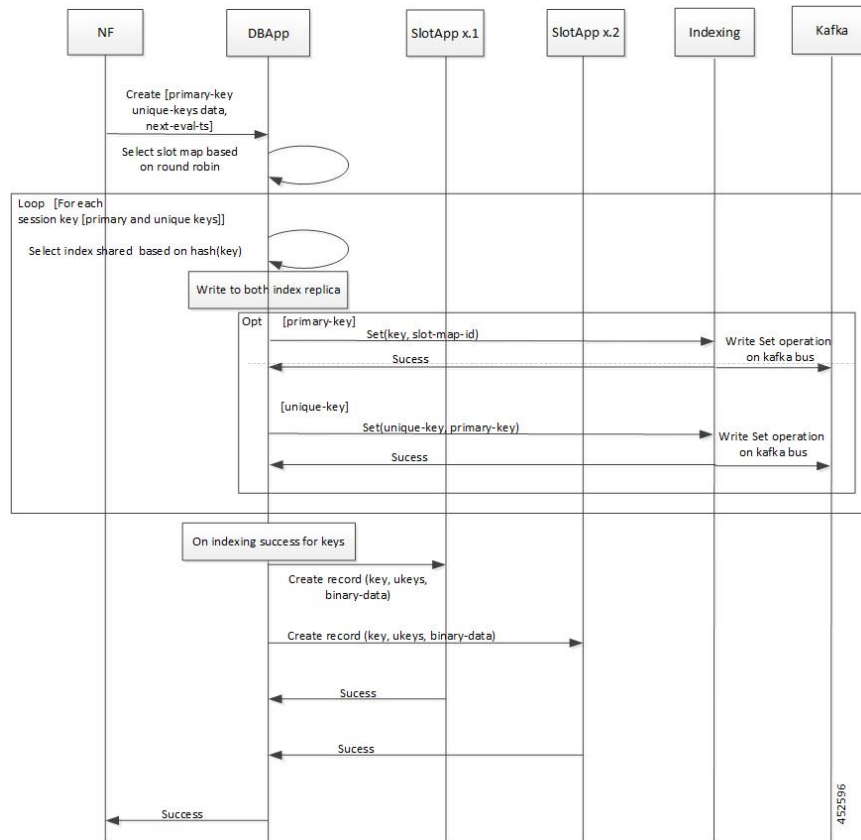
図 12:一意のキーによるレコードの検索 : コールフロー



レコードの作成

このセクションでは、レコードを作成するコールフローについて説明します。

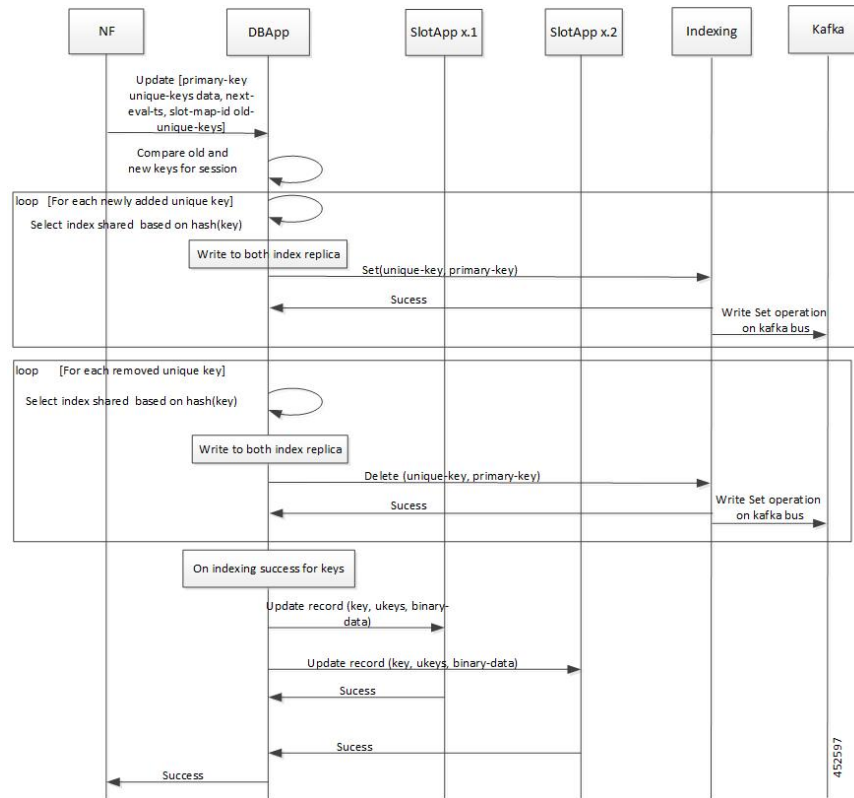
図 13: レコードの作成 : コールフロー



レコードの更新

このセクションでは、レコードの更新のコールフローについて説明します。

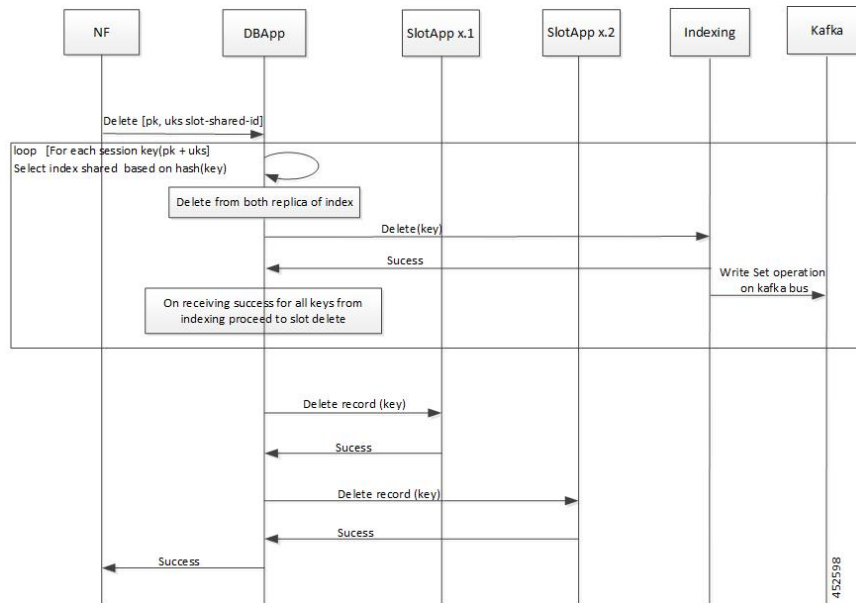
図 14: レコードの更新 : コールフロー



レコードの削除

このセクションでは、レコードの削除のコールフローについて説明します。

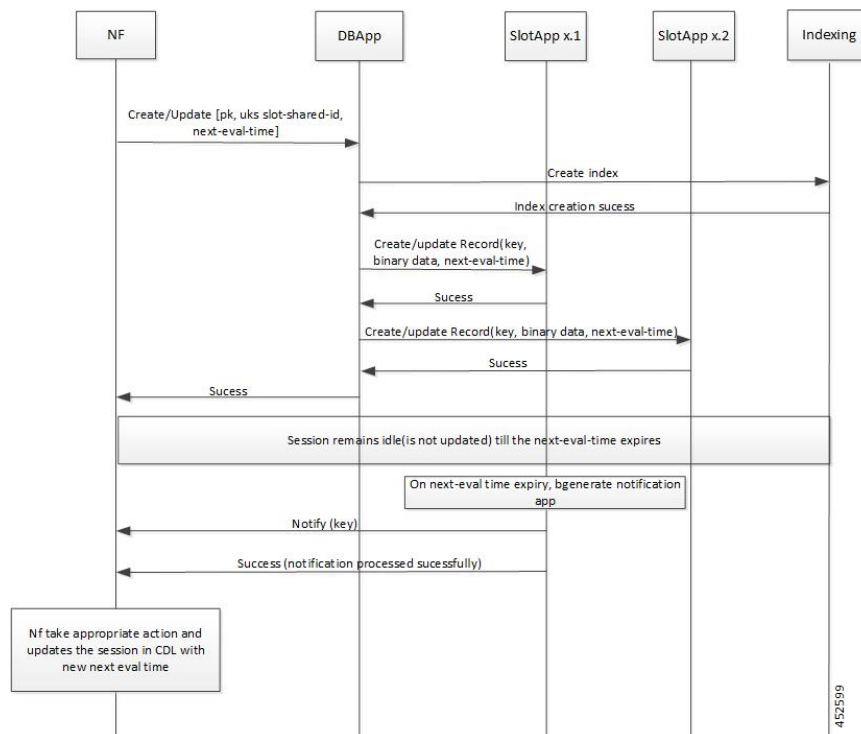
図 15: レコードの削除 : コールフロー



タイマーが切れた時の NF への通知

このセクションでは、タイマーが切れた時の NF へのセッション通知のコールフローについて説明します。

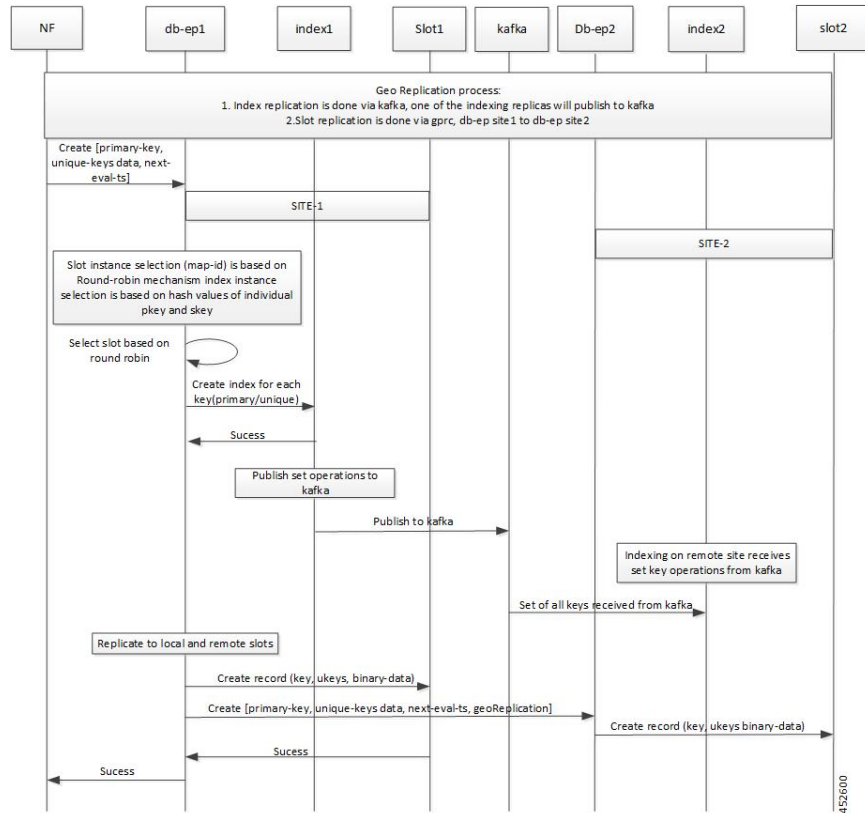
図 16: NF へのセッション通知 : コールフロー



Geo レプリケーション : 作成

このセクションでは、Geo レプリケーションのプライマリキーと一意のキーを作成するコールフローについて説明します。

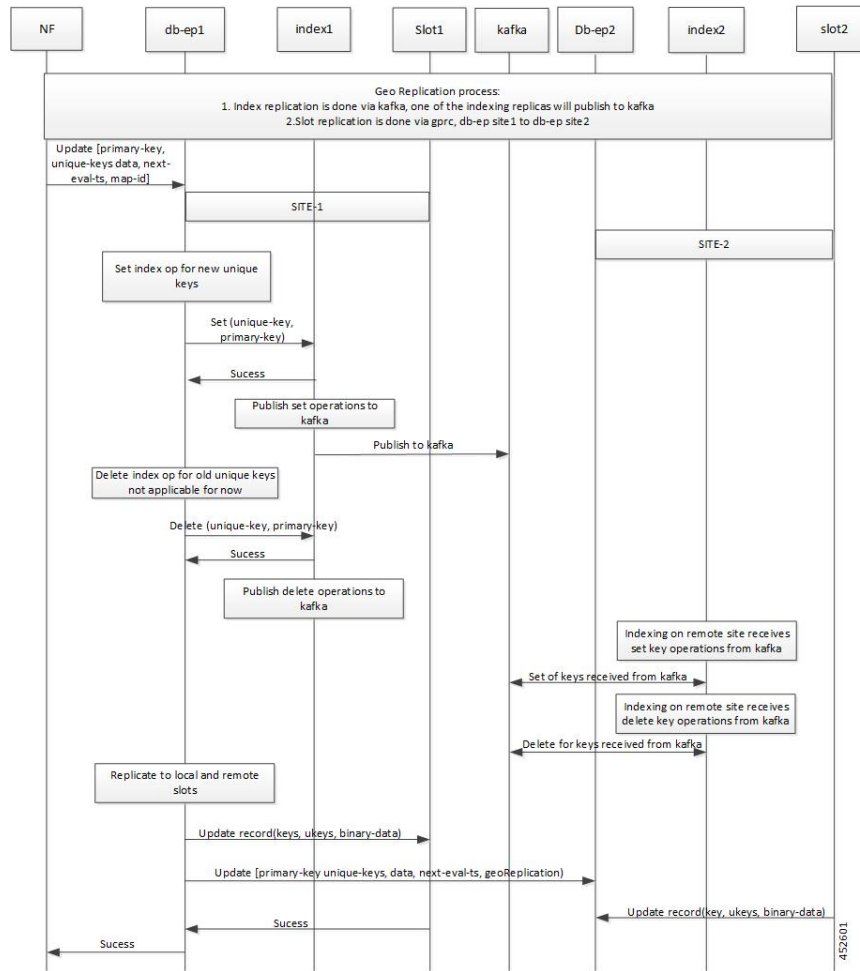
図 17: Geo レプリケーションのコールフロー : Geo の作成



Geo レプリケーション : 更新

このセクションでは、Geo レプリケーションのプライマリキーと一意のキーを更新するコールフローについて説明します。

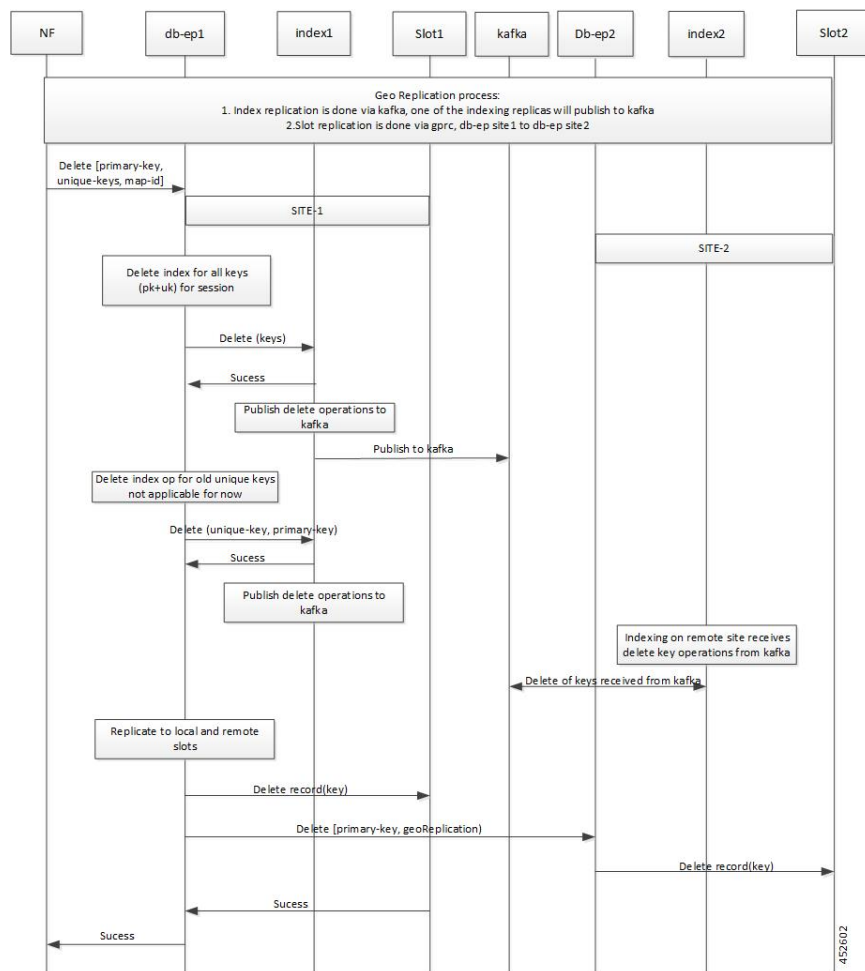
図 18: Geo レプリケーションのコールフロー：Geo の更新



Geo レプリケーション：削除

このセクションでは、Geo レプリケーションのプライマリキーと一意のキーを削除するコールフローについて説明します。

図 19: Geo レプリケーションのコールフロー : Geo の削除



古いインデックスレコードの識別

特定のシナリオでは、CDLのインデックスポッドの一意のキーが古くなっています（削除されると想定）。CDLは古いレコードの詳細をNFに示さないため、NFはこれらの一意のキーを別のレコードに使用しようとする場合があります。

CDLではオペレータが、古いレコードを検出し、その古いレコードに対して必要なアクションを実行するパラメータを有効にできます。

古いレコードを検出してアクションを実行するには、次の手順を実行します。

1. 古いレコードを識別します。新しいレコードが作成されると、一意のキーは上書きされます。このようなレコードを識別するには、**index-overwrite-detection** パラメータを有効にし、**unique-keys-prefix** がプレフィックスパターンと一致することを確認します。
2. 識別された古いレコードに対して必要な**アクション**（通知、削除、またはログ）を実行します。

CDL は、上書きされた一意のキーを検出し、次のいずれかのアクションを実行します。

- 古いレコードまたはレコード全体を削除します。削除アクションは、古いレコードの **PurgeOnEval** フラグが **false** に設定されている場合にのみトリガーされます。
- 古いレコードについて NF に通知します。通知アクションは、**STALE_INDEX_NOTIFICATION** を NF に送信します。
- 上書きされた一意のキーをログに記録します。ログアクションの場合、古いレコードは **WARN** ログレベルでログに記録されます。



- (注) 同じプライマリキーを指している2つの異なる一意のキーがあり、一方が通知アクションで、もう一方が削除アクションである場合、通知アクションが実行されます。

次の設定を使用します。

```
features index-overwrite-detection max-tps variable
features index-overwrite-detection queue-size variable
features index-overwrite-detection unique-keys-prefix uk
action [delete-record, notify-record, log-record]
```

値は次のとおりです。

- **max-tps** : 古いレコードの通知レートを制御します。デフォルトは 200 です。
- **queue-size** : 古いレコードを処理するキューサイズを制御します。デフォルトは 1000 です。



- (注) **queue-size** パラメータは、削除アクションと通知アクションの両方に使用されます。**max-tps** パラメータは通知アクション専用です。

- **unique-keys-prefix** : 実行する必要があるアクションとともに固有のキーのプレフィックスパターンを指定します。

例 :

```
cdl datastore session
features index-overwrite-detection max-tps 250
features index-overwrite-detection queue-size 2000
features index-overwrite-detection unique-keys-prefix uk
action notify-record
exit
exit
```

トラブルシューティング

古いインデックスレコードをトラブルシュートするには、**index.overwrite.session** ロガーを INFO レベルに設定します。エンドポイントポッドおよびインデックスポッドからのログは、障害対応に役立ちます。

CDL の設定：

```
cdl logging logger index.overwrite.session
  level info
exit
```

次のメトリックスが導入されています。

- **overwritten_index_records_deleted**：インデックスで特定された古いレコードが原因で削除されたレコードの総数を保持します。
- **overwritten_index_records_skipped**：古いレコードとして検出されたものの、そのレコードの通知または削除の処理中にキューがいっぱいになった場合はドロップされるレコードの総数を保持します。

ネットワーク機能（NF）を介した CDL の展開

ネットワーク機能（NF）（セッション管理機能（SMF）およびポリシー制御機能（PCF））の Ops Center を介して CDL を展開できます。

SMF Ops Center を介した CDL の展開については、『*Ultra Cloud Core 5G Session Management Function - Configuration and Administration Guide*』の「*Cisco Common Data Layer in SMF*」の章を参照してください。

PCF Ops Center を介した CDL の展開については、『*Ultra Cloud Core 5G Policy Control Function - Configuration and Administration Guide*』の「*Cisco Common Data Layer in PCF*」の章を参照してください。

CDL Geo レプリケーション（GR）の展開

このセクションでは、Geo レプリケーション（GR）の CDL を展開する方法について説明します。

CDL GR の前提条件

CDL GR を展開する前に、次の設定を行います。

- CDL セッションデータベースを設定し、基本設定を定義します。
- CDL 用の Kafka を設定します。
- CDL 用の Zookeeper を設定します。

CDL セッションデータベースの設定と基本設定の定義

このセクションでは、CDL セッションデータベースを設定し、NF（SMF または PCF）Ops Center を介して基本構成を定義する方法について説明します。

1. NF Ops Center コンソールを開き、データストア CLI に移動します。
2. セッションデータベースを設定し、CDL 操作のフェールオーバーの基本設定を定義するには、次の設定を使用します。

```
configure
cdl system-id system_id
cdl node-type node_type
cdl enable-geo-replication boolean_value
cdl remote-site remote_system_id db-endpoint host host_ip_address
db-endpoint port port
kafka-server remote_kafka_host remote_port
end exit

cdl datastore session
endpoint replica num_replica
endpoint external-ip ip_address
endpoint external-ip port
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
end exit
```

注：

- **cdl system-id system_id**：システムまたは Kubernetes クラスターの ID を指定します。デフォルト値は 1 です。
- **cdl node-type node_type**：ノードアフィニティを設定するための Kubernetes ノードラベルを示します。デフォルト値は session です。node_type は 0 ～ 64 文字の文字列である必要があります。
- **cdl enable-geo-replication boolean_value**：これはオプションの CLI です。Geo レプリケーションのステータスを有効または無効に指定します。デフォルト値は *false* です。
- **cdl remote-site remote_system_id**：リモート サイト エンドポイントのエンドポイントの IP アドレスを指定します。このコマンドは **cdl enable-geo-replication** を *true* に設定した場合にのみ設定します。
- **db-endpoint host host_ip_address**：リモートサイトのエンドポイント IP アドレスを指定します。このコマンドは **cdl enable-geo-replication** を *true* に設定した場合にのみ設定します。

- **db-endpoint port** *port_number* : リモート サイト エンドポイントのエンドポイントポートを示します。デフォルトのポート番号は 8882 です。このコマンドは **cdl enable-geo-replication** を *true* に設定した場合にのみ設定します。
- **kafka-server remote_kafka_host remote_port** : *remote-system-id* が識別する、リモートサイトの Kafka サーバーの外部 IP とポートを示します。リモートサイトの Kafka インスタンスごとに複数のホストとポートを設定できます。このコマンドは **cdl enable-geo-replication** を *true* に設定した場合にのみ設定します。
- **endpoint replica num_replica** : 作成されるレプリカの数を示します。デフォルト値は 1 です。*num_replica* は 1 ～ 16 の範囲内にする必要があります。
- **endpoint external-ip ip_address** : これはオプションの CLI です。データベースエンドポイントを公開する外部 IP アドレスを指定します。このコマンドは **cdl enable-geo-replication** を *true* に設定した場合にのみ設定します。
- **endpoint external-port port** : データベースエンドポイントを公開する外部ポートを指定します。このコマンドは **cdl enable-geo-replication** を *true* に設定した場合にのみ設定します。デフォルト値は 8882 です。
- **slot replica num_replica** : 作成するレプリカ数を指定します。デフォルト値は 1 です。*num_replica* は 1 ～ 16 の範囲内にする必要があります。
- **slot map num_map/shards** : スロット内のパーティション数を指定します。デフォルト値は 1 です。*num_map/shards* は 1 ～ 1024 の範囲内にする必要があります。
- **slot write-factor write_factor** : 正常な応答の前に書き込まれるコピーの数を指定します。デフォルト値は 1 です。*write_factor* は 0 ～ 16 の範囲内にする必要があります。値がレプリカ数以下であることを確認してください。
- **slot notification host host** : 通知サーバーのホスト名または IP アドレスを指定します。デフォルト値は *datastore-notification-ep* です。
- **slot notification port port** : 通知サーバーのポート番号を指定します。デフォルト値は 8890 です。
- **slot notification limit tps** : 1 秒あたりの通知上限を指定します。デフォルト値は 2000 です。
- **index replica num_replica** : 作成するレプリカ数を指定します。デフォルト値は 2 です。*num_replica* は 1 ～ 16 の範囲内にする必要があります。
- **index map num_map/shards** : スロット内のパーティション数を指定します。デフォルト値は 1 です。指定できる範囲は 1 ～ 1024 です。CDL の展開後は、この値を変更しないでください。
- **index write-factor write_factor** : 正常な応答の前に書き込まれるコピーの数を指定します。デフォルト値は 1 です。*write_factor* は 0 ～ 16 の範囲内にする必要があります。

CDL 用の Kafka の設定

このセクションでは、CDL 用の Kafka の設定方法を説明します。

1. NF Ops Center のコンソールを開き、データストア CLI に移動します。
2. 次の設定を使用します。

```
configure
  cdl kafka replica num_replicas
    enable-JMX-metrics boolean_value
    external-ip ip_address port_number
    retention-time retention_period
    retention-size retention_size
  end
exit
```



(注)

- **cdl kafka replica num_replicas** : 作成するレプリカの数指定します。デフォルト値は 3 です。num_replicas は 1 ～ 16 の範囲内である必要があります。
- **enable-JMX-metrics boolean_value** : JMX メトリックのステータスを指定します。デフォルト値は true です。
- **external-ip ip_address port_number** : Kafka サービスに公開する外部 IP を指定します。
enable-geo-replication パラメータを *true* に設定した場合に、このコマンドを設定します。Kafka レプリカの各インスタンスに外部 IP アドレスとポート番号を定義する必要があります。たとえば、**cdl kafka replica** パラメータが 3 に設定されている場合は、3 つの外部 IP アドレスとポート番号を定義する必要があります。
- **retention-time retention_period** : データを保持する必要がある期間 (時間単位) を指定します。デフォルト値は 3 です。retention_period は 1 ～ 168 の範囲内である必要があります。
- **retention-size retention_size** : データの保持サイズを MB 単位で指定します。デフォルト値は 5120 MB です。

CDL 用の Zookeeper の設定

このセクションでは、CDL 用の Zookeeper の設定方法を説明します。

CDL 用の Zookeeper を設定するには、次の設定を使用します。

1. NF Ops Center のコンソールを開き、データストア CLI に移動します。
2. 次のコマンドを実行します

```
configure
  cdl zookeeper data_storage_size_in_gb
  log-storage-size size_in_gb
  replica num_replicas
  enable-JMX-metrics boolean_value
  enable-persistence boolean_value
end
exit
```



(注)

- **cdl zookeeper data_storage_size_in_gb** : Zookeeper データストレージのサイズを GB 単位で指定します。デフォルト値は 20 GB です。data_storage_size_in_gb は 1 ～ 64 の範囲内である必要があります。
- **log-storage-size size_in_gb** : Zookeeper データログのストレージのサイズを GB 単位で示します。デフォルト値は 20 GB です。size_in_gb は 1 ～ 64 の範囲内である必要があります。
- **replica num_replicas** : 作成する必要があるレプリカの数を示します。デフォルト値は 3 です。num_replicas は 1 ～ 16 の範囲内である必要があります。
- **enable-JMX-metrics boolean_value** : JMX メトリックのステータスを示します。デフォルト値は true に設定されています。
- **enable-persistence boolean_value** : Zookeeper データの永続ストレージのステータスを指定します。デフォルト値は false です。

Geo レプリケーション (GR) 用の CDL の展開

Geo レプリケーション (GR) では、2 つの独立した HA システムを作成し、リモートサイトとの通信用に Geo HA を設定する必要があります。デフォルトでは、CDL は *db-ep* に 2 つのレプリカ、1 つのスロットマップ (マップごとに 2 つのレプリカ)、および 1 つのインデックスマップ (マップごとに 2 つのレプリカ) で展開されます。



重要

- YANG モデルで CDL コンテナを設定することをお勧めします。
- Geo HA は、2 つの HA システムがアクティブになってから設定することをお勧めします。

CDL GR を展開するには、次の設定を使用します。

手順

ステップ 1 サイト 1 に HA を展開します。

Geo HA 関連の設定パラメータを使用せずにサイト 1 の HA システムを展開し、設定で **system-id** パラメータを 1 に設定します。

```
configure
cdl system-id system_id
cdl node-type node_label
cdl datastore datastore_name
endpoint replica number_of_HA_instances
index map number_of_index_partitions
index replica number_of_HA_instances
```

```

slot map number_of_slot_partitions
slot replica number_of_HA_instances
exit

```

例 :

```

cdl# configure terminal
cdl(config)# cdl system-id 2
cdl(config)# cdl node-type session
cdl(config)# cdl datastore session
cdl(config-datastore-session)# endpoint replica 2
cdl(config-datastore-session)# slot map 4
cdl(config-datastore-session)# slot replica 2
cdl(config-datastore-session)# index map 3
cdl(config-datastore-session)# index replica 2
cdl(config-datastore-session)# exit
cdl(config)#

```

ステップ2 サイト1にHA設定を適用します。

設定をコミットし、サイト1にポッドを展開します。

ステップ3 サイト2にHAを展開します。

GeoHA関連の設定パラメータを使用せずにサイト2のHAシステムを（並行して）展開し、設定で**system-id**パラメータを2に設定します。

```

configure
  cdl system-id system_id
  cdl node-type node_label
  cdl datastore datastore_name
  endpoint replica number_of_HA_instances
  index map number_of_index_partitions
  index replica number_of_HA_instances
  slot map number_of_slot_partitions
  slot replica number_of_HA_instances
  exit

```

例 :

```

cdl# configure terminal
cdl(config)# cdl system-id 2
cdl(config)# cdl node-type session
cdl(config)# cdl datastore session
cdl(config-datastore-session)# endpoint replica 2
cdl(config-datastore-session)# slot map 4
cdl(config-datastore-session)# slot replica 2
cdl(config-datastore-session)# index map 3
cdl(config-datastore-session)# index replica 2
cdl(config-datastore-session)# exit
cdl(config)#

```

ステップ4 サイト2にHA設定を適用します。

設定をコミットし、サイト2にポッドを展開します。

ステップ5 サイト1とサイト2がアクティブであるかどうかを確認します。

Geo HA 設定に進む前に、CDL、Kafka、および Zookeeper ポッドがアクティブであるかどうかを確認します。

ステップ 6 サイト 2 の通信用にサイト 1 で外部 IP アドレスを設定します。

- a) サイト 1 で Geo レプリケーションを有効にし、リモートサイトをサイト 1 の 2 として設定します。

```
configure
  cdl enable-geo-replication true/false
  cdl datastore session geo-remote-site list_of_geo_replication_sites
```

例：

```
cdl# configure terminal
cdl(config)# cdl enable-geo-replication true
cdl(config)# cdl datastore session geo-remote-site 2
```

- b) CDL エンドポイントの外部 IP アドレスを、サイト 2 がアクセスできるように設定します。

```
configure
  cdl datastore session endpoint external-ip site_1_external_ip_address
```

- c) すべての Kafka レプリカの外部 IP アドレスとポートを設定します。

たとえば、Kafka に 2 つのレプリカ（デフォルト）が設定されている場合、IP アドレスとポート番号の 2 つの異なるペアを指定します。

```
configure
  cdl kafka external-ip site_1_external_ip_address port1
  cdl kafka external-ip site_1_external_ip_address port2
```

- d) (オプション) SSL/TLS 証明書を設定して、ローカルサイトの TLS サポートを有効にします。

TLS 証明書を設定すると、ローカルサイトでセキュアな接続と非セキュアな接続の両方を受け入れることができます。同様に、SSL 証明書を設定すると、ローカルサイトはリモートサイトとのセキュアな接続を確立できます。

(注)

証明書はすべて、複数行の生のテキスト形式です。証明書が無効な場合、サーバーは非セキュアな接続を続行します。

```
configure
  cdl ssl-config enable true
  cdl ssl-config ip site_1_external_ip_address
  cdl ssl-config certs site_1_external_ip_address ssl-key ssl_key
  cdl ssl-config certs site_1_external_ip_address ssl-crt ssl_crt
```

ステップ 7 サイト 1 の通信用にサイト 2 に外部 IP アドレスを設定します。

- a) サイト 2 で Geo レプリケーションを有効にし、リモートサイトをサイト 2 の 1 として設定します。

```
configure
  cdl enable-geo-replication true/false
  cdl datastore session geo-remote-site list_of_geo_replication_sites
```

例：

```
cdl# configure terminal
cdl(config)# cdl enable-geo-replication true
cdl(config)# cdl datastore session geo-remote-site 1
```

- b) CDL エンドポイントの外部 IP アドレスを、サイト 1 がアクセスできるように設定します。

```
configure
cdl datastore session endpoint external-ip site_2_external_ip_address
```

- c) すべての Kafka レプリカの外部 IP アドレスとポートを設定します。

たとえば、Kafka に 2 つのレプリカ（デフォルト）が設定されている場合、IP アドレスとポート番号の 2 つの異なるペアを指定します。

```
configure
cdl kafka external-ip site_2_external_ip_address port1
cdl kafka external-ip site_2_external_ip_address port2
```

- d) （オプション）SSL/TLS 証明書を設定して、ローカルサイトの TLS サポートを有効にします。

TLS 証明書を設定すると、ローカルサイトでセキュアな接続と非セキュアな接続の両方を受け入れることができます。同様に、SSL 証明書を設定すると、ローカルサイトはリモートサイトとのセキュアな接続を確立できます。

（注）

証明書はすべて、複数行の生のテキスト形式です。証明書が無効な場合、サーバーは非セキュアな接続を続行します。

```
configure
cdl ssl-config enable true
cdl ssl-config ip site_1_external_ip_address
cdl ssl-config certs site_1_external_ip_address ssl-key ssl_key
cdl ssl-config certs site_1_external_ip_address ssl-crt ssl_crt
```

ステップ 8 各サイトにリモートサイト情報を追加します。

- a) サイト 1 でリモートサイトの *cdl-ep*（CDL エンドポイント）の設定を行います。

```
configure
cdl remote-site 2 db-endpoint host site_2_cdl_ep_ip
```

- b) サイト 1 でリモートサイトの Kafka の設定を行います。

```
configure
cdl remote-site 2 kafka-server site_2_kafka1_ip_address site_2_kafka1_port
cdl remote-site 2 kafka-server site_2_kafka2_ip_address site_2_kafka2_port
```

- c) （オプション）SSL 証明書を設定して、サイト 1 でリモートサイトとのセキュアな接続を確立します。

（注）

証明書はすべて、複数行の生のテキスト形式です。証明書が無効な場合、サーバーは非セキュアな接続を続行します。

```
configure
cdl ssl-config certs site_2_external_ip_address ssl-key ssl_key
cdl ssl-config certs site_2_external_ip_address ssl-crt ssl_crt
```

サイト 1 でその設定をコミットします。

- d) サイト 2 でリモートサイトの `cdl-ep` の設定を行います。

configure

```
remote-site 1 db-endpoint host site_1_cdl_ep_ip_address
```

- e) サイト 2 でリモートサイトの `Kafka` の設定を行います。

configure

```
cdl remote-site 1 kafka-server site_1_kafka1_ip_address site_1_kafka1_port
```

```
cdl remote-site 1 kafka-server site_1_kafka2_ip_address site_1_kafka2_port
```

- f) (オプション) SSL 証明書を設定して、サイト 2 でリモートサイトとのセキュアな接続を確立します。

(注)

証明書はすべて、複数行の生のテキスト形式です。証明書が無効な場合、サーバーは非セキュアな接続を続行します。

configure

```
cdl ssl-config certs site_1_external_ip_address ssl-key ssl_key
```

```
cdl ssl-config certs site_1_external_ip_address ssl-crt ssl_crt
```

サイト 2 でその設定をコミットします。

ステップ 9 Mirror Maker ポッドが展開されているかどうか、および他のすべてのポッドがアクティブであるかどうかを確認します。

注：

- **cdl kafka external-ip** : すべての `Kafka` レプリカの外部 IP アドレスとポートを指定します。
 - `site_external_ip_address` : リモートサイトの外部 IP アドレスを指定します。
 - `port_number` : リモートサイトのポート番号を指定します。
- **cdl ssl-config certs** : SSL 証明書を指定して、リモートサイトとのセキュアな接続を確立します。
 - `site_external_ip_address` : リモートサイトの外部 IP アドレスを指定します。
 - `ssl-keyssl_key` : 生成された SSL キーを指定します。

Geo レプリケーション (GR) のフェールオーバーの通知

CDL には Geo レプリケーション (GR) フェールオーバー通知が用意されているため、セッションデータのタイマー切れの通知や、現在アクティブなサイトへの一括通知が可能です。CDL は GR フェールオーバー通知のために、App-Infra を介してボーダー ゲートウェイ プロトコル (BGP) を使用します。

CDL は両方の GR サイトでキー値を登録します。登録したキー値に変更があると、App-Infra は CDL に通知を送信します。キー値は、CDL システム ID または GR インスタンスの状態を示

します。GR インスタンスは、キーの CDL システム ID または GR インスタンス ID を使用して CDL スライスにマップされます。

CDL GR フェールオーバー通知に、次のパラメータが導入されました。

- **instance-aware-notification enable true or false** : GR フェールオーバー通知を有効にするには、これを true に設定します。
- **instance-aware-notification system-id** : system-id は sliceName にマップされます。つまり、プライマリ system-id はその特定の sliceName のプライマリサイト ID にマップされます。この情報は、すべての Geo サイトで構成する必要があります。FindAllNotify 通知および TimerExpiry 通知は、一括処理タスクのシステム ID の詳細を使用します。

システム ID は両方のサイトで必須です。NF 構成の GR インスタンス ID が CDL システム ID と一致している必要があります。

次の例では、GR フェールオーバー通知が true に設定されています。system-id 1 は slicesName *sgw1* および *smf1* のプライマリサイト ID で、system-id 2 は slicesName *sgw2* および *smf1* のプライマリサイトです。

```
cdl datastore session
features instance-aware-notification enable true
features instance-aware-notification system-id 1
  slice-names [ sgw1 smf1 ]
exit
features instance-aware-notification system-id 2
  slice-names [ sgw2 smf2 ]
exit
exit
```

ピアサイトの GR フェールオーバー通知レコード

CDL は、ローカルアプリケーションにのみレコードに関する通知を送信します。通知は、レコードの system-id パラメータと Timer Expiry パラメータに基づいています。GR セットアップでは、メンテナンスのためにサイトが分離されると、そのサイトのレコードはピアサイトに送信されません。

remote-system-id パラメータを使用すると、CDL はピアサイトが分離されたサイトの通知を処理することを許可します。ピアサイトの **remote-system-id** は、分離されたサイトのサイト ID で設定されます。レコードの system-id が remote-system-id と一致する場合、CDL はレコードを処理します。通知は、Timer Expiry に基づいて、または **notifyOnPurge** が有効になっているレコードに関して送信されます。



- (注) 分離されたサイトの機能が復帰した後は、その remote-system-id を CDL 設定から削除する必要があります。

次の手順では、例を示してこの remote-system-id の設定について説明します。

次の例では、GR セットアップに 2 つのサイト（サイト 1 とサイト 2）があります。サイト 1 はアップグレードのために切断され、サイト 2 の remote-system-id がサイト 1 のサイト ID で設定されます。

1. サイト 1 をシャットダウンまたは切断します。
2. サイト 1 のサイト ID を使用してサイト 2 の **remote-system-id** を設定するには、次のコマンドを実行します。

```
cdl datastore session
  slot notification remote-system-id [ 1 ]
exit
```

上記のコマンドの **remote-system-id** の値 [1] は、分離されているサイト 1 のサイト ID であることに注意してください。

3. サイト 2 は、ローカルアプリケーションへのサイト 1 のレコードの通知を開始します。
4. サイト 1 を起動する前に、サイト 2 の **remote-system-id** リストからサイト 1 のサイト ID を削除します。

remote-system-id は、**instance-aware-notification-system-id** と相互に排他的です。詳細については、「*Geo* レプリケーション (GR) のフェールオーバーの通知」のトピックを参照してください。

リモートインデックス同期のトリガー

CDL は、インデックスをリモートピアと同期させるためのユーティリティを提供します。このユーティリティは、サイト間でインデックスレコードの数に大きな違いがある場合の、サイト分離後などのシナリオで使用できます。CDL は、リモートインデックス同期のステータスを確認するコマンドもサポートしています。



- (注)
- パフォーマンスに影響を与える可能性があるため、リモート同期はサイト間のインデックスレコードに大きな違いがある場合にのみ使用することをお勧めします。
 - コマンドを表示するには、**geo-remote-site** を設定する必要があります。

以下に関するリモートインデックスの同期をトリガーします。

- インデックス **mapID** のリスト、または CLI からのすべての **mapID**。
- **sliceName** のリスト、またはすべての **sliceName**。

次の点に注意してください。

- インデックスのピアとのリモート同期のトリガー中のエラーを回避するために、次の条件を満たしていることを確認してください。
 - カスタムの **map-id** と **slice-name** は一意にする必要があります。
 - 有効なインデックス **map-id** と **slice-name** のみが許可されます。
 - リモートサイトは到達可能です。

- 内部的に、CDL はリモートピアとの同期のために最大 3 回再試行します。
- インデックスインスタンスの同期がすでに進行中の場合、リモートインデックス同期はそのインスタンスをスキップします。

コマンド	出力パラメータ
cdl actions remote-index-sync start <i>[options]</i> [オプション (Options)] • map-id リモートインデックス同期を開始する必要があるインデックス map-id。map-id はオプションです。含めると、そのインデックス map-id のすべてのインスタンスのリモートインデックス同期がトリガーされます。このオプションを使用して、最大 5 つの map-id を指定できます。 • slice-name リモートインデックス同期を開始する slice-name。slice-name はオプションです。含めると、すべての slice-name のリモートインデックス同期がトリガーされます。slice-name の数に制限はありません。	triggered-instances リモートインデックス同期が開始されているインデックスインスタンスのリストを表示します。
例 cdl actions remote-index-sync start map-id { 1 } map-id { 2 } slice-name { session-1 } slice-name { session-2 }	出力 triggered-instances 'index-mapID-1-instanceID-1, index-mapID-1-instanceID-2, index-mapID-2-instanceID-1, index-m

リモート同期ステータスの確認

リモート同期ステータスには、リモートインデックス同期が進行中のインデックスインスタンスが表示されます。



- (注) リモート同期ステータスには、sliceName ごとの同期ステータスは表示されません。同期ステータスに特定のインデックスインスタンスがリモートサイトと同期していることが示されている場合は、すべての sliceName またはリストされている sliceName が順番に同期していることを意味します。

コマンド cdl actions remote-index-sync status	出力パラメータ syncing-instances リモートインデックス同期が進行中のインデックスインスタンスのリスト。
例 cdl actions remote-index-sync status	出力 syncing-instances 'index-mapID-1-instanceID-1, index-mapID-1-instanceID-2, index-mapID-2-instanceID-1, index-mapID-2-instanceID-2'

トラブルシューティング

インデックスポッドの次の警告またはエラーログには、リモートインデックス同期のステータスが表示されます。

- 同期が成功している場合は、次のログが表示されます。

ログ：

Bulk Sync done from Remote Indexes(s) for isInitSync = false

例：

```
[datastore.index.session] Bulk Sync done from Remote Indexes(s) for isInitSync = false sliceName = session via DBApp
```

ログ：

Sync finished successfully with Remote MemoryMap App isInitSync: false

例：

```
[datastore.index.session] Sync finished successfully with Remote MemoryMap App isInitSync: false count: 100 Time taken: 16.699031ms sliceName: session
```

- 同期が失敗している場合は、次のエラーログが表示されます。

例：

```
Error! Remote Bulk Read failed 3 times for isInitSync = false
```

機能の概要と変更履歴

要約データ

該当製品または機能エリア	PCF 2021.04.0 以降
該当するプラットフォーム	ベアメタル、OpenStack、VMware
機能のデフォルト設定	無効：設定が必要

このリリースでの関連する変更点	N/A
関連資料	該当なし

マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	CDL 1.6

機能説明

このリリースでは、CDLはIPv6をサポートしているため、GRが有効になっているセットアップではエンドポイントのデュアルスタックサポートが有効になります。デュアルスタックにより、IPv4 アドレスと IPv6 アドレスの両方を使用してネットワーキングデバイスを設定できます。

この機能により、次のような機能が提供されます。

- CDL エンドポイントの IPv6 サポート。
- Mirror Maker と Kafka ブローカ間の通信用の IPv6 サポート。
- CDL および Kafka IPv6 エンドポイントの TLS サポート。



- (注) CDL で IPv6 サポートを行うには、K8s クラスタがデュアルスタックをサポートしており、それがクラスタで有効になっている必要があります。

IPv6 サポートの設定

GR が有効になっているセットアップで CDL の IPv6 サポートを設定するには、次の手順で CLI コマンドまたは設定を使用します。

1. CDL IPv6 エンドポイントを公開し、リモートサイトの IPv6 エンドポイントに接続します。

1. CDL Geo レプリケーションを有効にします。

```
config
cdl enable-geo-replication true
```

2. CDL エンドポイントの外部 IP アドレスを設定します。**endpoint external-ip** に加えて、**endpoint external-ipv6** パラメータも設定します。

```
config
cdl datastore session endpoint external-ipv6 IPv6_address
```

3. リモートサイトの CDL エンドポイントを設定します。

```
config
cdl remote-site system_id
```

```
db-endpoint host IPv6_address
db-endpoint port 8882
exit
```



- (注) IPv4 エンドポイントと IPv6 エンドポイントには、同じポート（デフォルト値は 8882）を介してアクセスできます。



- (注) 現在、CDL は IPv4 または IPv6 のいずれかを使用してリモートサイトのエンドポイントに接続できますが、両方を使用することはできません。ただし、リモートサイトの IPv4 エンドポイントと IPv6 エンドポイントの両方を同時に公開して、いずれかに接続することはできます。

2. CDL Kafka を公開し、IPv6 を介してリモートサイトの Kafka に接続します。

- すべての Kafka レプリカの外部 IP アドレスとポートを設定します。

```
config
cdl kafka external-ipv6 IPv6_address port_address
exit
```



- (注) 各ブローカに個別に IPv6 を設定する必要があります。IPv4 と IPv6 を介して Kafka ブローカを同時に公開することは可能ですが、それぞれ異なるポートを使用する必要があります。Kafka リスナーは現在、異なる IP アドレスに対する同じポートの使用をサポートしていません。

- リモートサイトの Kafka 設定を行います。

Mirror Maker は、IPv4 と IPv6 の両方またはいずれか 1 つを介して、ローカルサイトとリモートサイトの両方の Kafka ブローカに接続できます。

次の CLI コマンドを使用して、リモートの Kafka サーバーを個別に設定します。

```
config
cdl remote-site system_id
kafka-server IPv6_address port_address
exit
```

- （オプションの手順）SSL/TLS 証明書を設定して、ローカルサイトとリモートサイトの両方で TLS サポートを有効にします。これらの証明書は、ローカルサイトとリモートサイト間のセキュアな接続を確立するのに役立ちます。

同じ設定を使用して、各サイトの証明書を設定します。

- CDL エンドポイントの SSL を有効にします。

```
config
cdl ssl-config enable true
```

2. 証明書を設定します。

```
config
cdl ssl-config certs external_ipv6_address
    ssl-key ssl_key
    ssl-crt ssl_crt
    ssl-ca ssl_ca
exit
```

注：

- **ssl-key** *ssl_key*：サーバーの秘密キーを指定します。
- **ssl-crt** *ssl_crt*：CA 署名付きサーバー証明書を指定します。
- **ssl-ca** *ssl_ca*：パブリック CA 証明書を指定します。

4. (オプションの手順) SSL/TLS 証明書を設定して、Kafka エンドポイントの TLS サポートを有効にします。

1. CDL エンドポイントの SSL を有効にします。

```
config
cdl ssl-config enable true
```

2. Kafka エンドポイントの SSL を有効にします。

```
config
cdl kafka ssl-settings enable-ssl true
cdl kafka ssl-settings disable-host-name-verification true
```

3. ローカルサイトとリモートサイトの両方で Kafka エンドポイントの証明書を設定します。

```
config
cdl ssl-config certs external_ipv6_address
    ssl-key ssl_key
    ssl-crt ssl_crt
    ssl-ca ssl_ca
exit
```

次の設定例は、リモートサイトの IPv6 TLS エンドポイントと IPv6/IPv4 TLS Kafka に接続するために使用します。

サイト 1 の設定

```
cdl remote-site 2
db-endpoint host 2001:DB8:54ff:a4::139:250
db-endpoint port 8882
kafka-server 10.106.139:250 10092
ssl-port 10094
exit
kafka-server 10.106.139:250 10093
ssl-port 10095
exit
kafka-server 2001:DB8:54ff:a4::139:250 10096
ssl-port 10098
```

```

exit
kafka-server 2001:DB8:54ff:a4::139:250 10097
ssl-port 10099
exit
exit
cdl ssl-config enable true
cdl ssl-config certs 2001:DB8:54ff:a4::139:250
    ssl-key "<server-key>"
    ssl-crt "<signed-certificate>"
    ssl-ca "<ca-certificate>"
exit
cdl ssl-config certs 192.0.2.2
    ssl-key "<server-key>"
    ssl-crt "<signed-certificate>"
    ssl-ca "<ca-certificate>"
exit
cdl ssl-config certs 192.0.2.1
    ssl-key "<server-key>"
    ssl-crt "<signed-certificate>"
    ssl-ca "<ca-certificate>"
exit
cdl ssl-config certs 2001:DB8:54ff:a4::139:249
    ssl-key "<server-key>"
    ssl-crt "<signed-certificate>"
    ssl-ca "<ca-certificate>"
exit
cdl datastore session
    geo-remote-site [ 2 ]
    endpoint external-ip 192.0.2.1 endpoint external-ipv6 2001:DB8:54ff:a4::139:249
exit
cdl kafka ssl-settings enable-ssl true
cdl kafka ssl-settings disable-host-name-verification true
cdl kafka external-ipv6 2001:DB8:54ff:a4::139:249 10096
    ssl-port 10098
exit
cdl kafka external-ipv6 2001:DB8:54ff:a4::139:249 10097
    ssl-port 10099
exit
cdl kafka external-ip 10.106.139.249 10092
    ssl-port 10094
exit
cdl kafka external-ip 10.106.139.249 10093
    ssl-port 10095
exit

```

サイト 2 の設定

```

cdl remote-site 1
db-endpoint host 2001:DB8:54ff:a4::139:249
db-endpoint port 8882
kafka-server 10.106.139.249 10092
ssl-port 10094
exit
kafka-server 10.106.139.249 10093
ssl-port 10095
exit
kafka-server 2001:DB8:54ff:a4::139:249 10096
ssl-port 10098
exit
kafka-server 2001:DB8:54ff:a4::139:249 10097
ssl-port 10099
exit
exit
cdl ssl-config enable true
cdl ssl-config certs 2001:DB8:54ff:a4::139:249

```

```

        ssl-key "<server-key>"
        ssl-crt "<signed-certificate>"
        ssl-ca "<ca-certificate>"
    exit
cdl ssl-config certs 192.0.2.1
    ssl-key "<server-key>"
    ssl-crt "<signed-certificate>"
    ssl-ca "<ca-certificate>"
exit
cdl ssl-config certs 192.0.2.1
    ssl-key "<server-key>"
    ssl-crt "<signed-certificate>"
    ssl-ca "<ca-certificate>"
exit
cdl ssl-config certs 2001:DB8:54ff:a4::139:250
    ssl-key "<server-key>"
    ssl-crt "<signed-certificate>"
    ssl-ca "<ca-certificate>"
exit
cdl datastore session
    geo-remote-site [ 1 ]
    endpoint external-ip 192.0.2.1
    endpoint external-ipv6 2001:DB8:54ff:a4::139:250
exit
cdl kafka ssl-settings enable-ssl true
cdl kafka ssl-settings disable-host-name-verification true
cdl kafka external-ipv6 2001:DB8:54ff:a4::139:250 10096
    ssl-port 10098
exit
cdl kafka external-ipv6 2001:DB8:54ff:a4::139:250 10097
    ssl-port 10099
exit
cdl kafka external-ip 192.0.2.1 10092
    ssl-port 10094
exit
cdl kafka external-ip 192.0.2.1 10093
    ssl-port 10095
exit

```

CDL アラーム

このセクションでは、CDL デプロイメントのアラームについて説明します。

障害を検出するために次のアラームが導入されています。

表 2: アラーム

アラーム	シビラティ (重大度)	説明
cdlLocalRequestFailure	critical	このアラームは、ローカル要求の成功率が 5 分以上にわたって 90% 未満の場合にトリガーされます。

アラーム	シビラティ (重大度)	説明
cdlRemoteConnectionFailure	critical	このアラームは、エンドポイントポッドからリモートサイトへのアクティブな接続が 5 分以上にわたって 0 に達した場合にトリガーされます。このアラームは GR デプロイメント専用です。
cdlRemoteRequestFailure	critical	このアラームは、受信リモート要求の成功率が 5 分以上にわたって 90% 未満の場合にトリガーされます。このアラームは GR デプロイメント専用です。
cdlReplicationError	critical	cdl-global 名前空間でのローカル要求に対する発信レプリケーション要求の比率が 5 分以上にわたって 90% 未満の場合。このアラームは GR デプロイメント専用です。
cdlKafkaRemoteReplicationDelay	critical	このアラームは、リモートサイトへの Kafka レプリケーションの遅延が 5 分以上にわたって 10 秒を超えた場合にトリガーされます。このアラームは GR デプロイメント専用です。

CDL ゾーンのアップグレード

機能説明

CNDPは、アップグレードゾーン内のすべてのノードが並行してアップグレードされる、ゾーンベースのアップグレードをサポートしています。これにより、大規模なクラスタのアップグレードに時間をかけずに済みます。CDLは、K8sのトポロジ分散制約機能を利用して、アップグレードゾーン全体にポッドを分散します。

CDLでこのインサースビスをサポートするには、ポッドがアップグレードゾーンを認識している必要があります。スロット、インデックス、またはKafkaのすべてのレプリカが同じアップグレードゾーンでスケジュールされないようにする必要があります。たとえば、同じマップの両方のスロットレプリカが同じアップグレードゾーンにある場合、両方のレプリカが同時にダウンし、非GRシナリオでセッションの損失を引き起こします。CDLエンドポイントは、レプリカが1つ戻るまで要求の処理を停止します。

少なくとも3つのアップグレードゾーンが存在することを確認してください。ノード障害が発生すると、別のゾーンにポッドをスケジュールするスペースがある場合でも、ポッドが保留状態になる可能性があります。

この機能を有効にするために CDL または NF Ops Center で必要な明示的な設定はありません。クラスタデプロイ自体が設定を渡します。

トポロジの分散制約

CDL は、K8s トポロジの分散制約機能を使用してアップグレードゾーン全体にポッドを分散し、アップグレードゾーンがダウンした場合にすべての冗長ポッドが同時に失われないようにします。

トポロジの分散制約について、次のフィールドが定義されています。

- **maxSkew** : ターゲットトポロジで一致するポッドの数とグローバル最小数（トポロジドメインのラベルセクタに一致するポッドの最小数）との間で許容される最大差を定義します。たとえば、それぞれ 2 つ、4 つ、5 つの一致するポッドを持つ 3 つのゾーンがある場合、グローバル最小数は 2 となり、**maxSkew** はその数に対して比較されます。
- **whenUnsatisfiable** : 分散制約を満たさない場合のポッドの処理方法を示します。
 - **DoNotSchedule** (デフォルト) は、スケジューラがポッドをスケジュールしないことを示します。
 - **ScheduleAnyway** は、スケジューラがスキューを最小限に抑えるノードに優先順位を付けながらポッドをスケジュールすることを示します。
- **topologyKey** は、ノードラベルのキーです。2 つのノードがこのキーでラベル付けされ、そのラベルの値が同じである場合、スケジューラは両方のノードを同じトポロジ内にあるものとして扱います。スケジューラは、バランスの取れた数のポッドを各トポロジドメインに配置しようとします。
- **labelSelector** を使用して、一致するポッドを検索します。このラベルセクタに一致するポッドをカウントし、対応するトポロジドメイン内のポッドの数を決定します。

CDL ポッドについて、トポロジ分散制約のために定義されたフィールドの例を次に示します。

- **topologyKey**: `smi.cisco.com/upgrade-zone`
- **maxSkew**: 1
- **whenUnsatisfiable**: `DoNotSchedule`
- **labelSelector**: デプロイメントまたはステートフルセットに一致するすべてのポッドを選択します。

K8s スケジューラはこの制約に基づき、**maxSkew** が 1 のゾーン全体へのデプロイメントまたはステートフルセットでポッドを均等に分散しようとします。

- 2 つのレプリカを持つスロットまたはインデックスポッドでは、2 つのレプリカは常に別々のアップグレードゾーンにあります。
- `cdl-ep` または `Kafka` ポッドの場合、ポッドは **maxSkew** が 1 の状態でほぼ均等に分散されます。

- ポッドが **maxSkew: 1** 制約を満たすことができない場合、**whenUnsatisfiable** フィールドが **DoNotSchedule** に設定されるため、そのポッドは保留状態のままになります。この制約は、**maxSkew** の違反を許可しません。



(注) トポロジの分散制約は、**cdl-ep**、**cdl-slot**、インデックス、**Kafka**、またはステートフルセットのデプロイメントにのみ追加されます。**etcd** と **zookeeper** は、1 つずつアップグレードされるプライマリノードまたは **OAM** ノードでスケジュール設定されるため、変更はありません。

制限事項と制約事項

このセクションでは、**CDL** ゾーンアップグレードの既知の制限事項と制約事項について説明します。

- **CDL** は、**Kafka** レプリカ 2 または **Kafka** レプリカ 3 を使用した **NF** のデプロイメントをサポートしています。**Kafka** レプリカ 3 を使用したデプロイメントの場合、**CDL** は少なくとも 3 つのアップグレードゾーンを定義します。3 つのレプリカと 2 つのアップグレードゾーンがある場合、一方のアップグレードゾーンに 2 つのレプリカがあり、もう一方のアップグレードゾーンに 1 つのレプリカがあります。2 つのレプリカを持つアップグレードゾーンがダウンしたこのシナリオでは、1 つのレプリカのみがトラフィックを処理できます。
- アップグレードゾーン内のノードがすべて使用されている場合、ゾーン内の 1 つ以上のノードがダウンすると、ダウンした 1 つ以上のノードのポッドが保留状態になる可能性があります。

次の条件は、同じゾーンノードと異なるゾーンノードに適用されます。

- 同じゾーンノード：同じゾーン内のノードに、そこでポッドをスケジュールするために要求された **CPU** またはメモリリソースがない。
- 異なるゾーンノード：次の条件のいずれかが異なるゾーンノードに適用されます。
 - 別のゾーン内のノードに、そこでポッドをスケジュールするために要求された **CPU** またはメモリリソースがない。
 - 別のゾーンのノードに、ポッドをスケジュールするために要求された **CPU** またはメモリリソースがある。スケジュール設定では、ポッドは **maxSkew:1** の制約に違反します。

CDL のオーバーロードの防止

CDL は、システムがフルキャパシティに達するたびに、エンドポイントでオーバーロードを防止します。エンドポイントでのオーバーロード防止により、**CDL** は次のことができます。

- ソフト制限の上限に達した場合、作成要求を拒否する。

- ハード制限の上限に達した場合、更新 要求を拒否する。
- システムが最大キャパシティの 80% および 90% に達したときにアラームをトリガーする。
- エンドポイントでオーバーロード防止を設定（有効または無効）する。

CDL の最大キャパシティ

次のパラメータを使用して、CDL のシャードあたりの最大キャパシティを定義できます。

- *slotRecordCapacity*（デフォルト値：1M/シャード）
- *indexRecordCapacity*（デフォルト値：10M/シャード）
- *slotRecordsSizeCapacityInBytes*（デフォルト値：5GB/シャード）



(注) シャードあたりのデフォルトのキャパシティは変更できません。

CDL は次の計算に基づいて最大キャパシティを決定します。

ソフト制限

1. `cdl_slot_record_capacity = No of shards * slotRecordCapacity`
2. `cdl_index_record_capacity = No of shards * indexRecordCapacity`
3. `cdl_slot_size_capacity = No of shards * slotRecordsSizeCapacityInBytes`

ハード制限

1. `cdl_slot_record_capacity = No of shards * slotRecordCapacity * 105%`
2. `cdl_index_record_capacity = No of shards * indexRecordCapacity * 105%`
3. `cdl_slot_size_capacity = No of shards * slotRecordsSizeCapacityInBytes * 105%`

CDL でのオーバーロードのアラートのトリガー

システムが最大キャパシティに達するたびに、CDL は過負荷となります。また、CDL は次のアラートをトリガーして、システムを過負荷から保護します。

- *cdlOverloaded - major*：このアラートは、CDL システムがキャパシティの 80% に達するとトリガーされます。
- *cdlOverloaded - critical*：このアラートは、CDL システムがキャパシティの 90% に達するとトリガーされます。

アラートに加え、CDL エンドポイントは次のエラーによってすべての作成要求を拒否します。

`error code 507: Datastore reached its full capacity`

ただしCDLは、ハード制限キャパシティに達するまで、すべての更新要求を処理します。ハード制限キャパシティを超えると、CDLはすべての更新要求を拒否します。

オーバーロードの防止の設定

既存のCDLのオーバーロード防止設定は、新しい設定に置き換えられます。



- (注) 以前はオーバーロードの防止がデフォルトで有効になっていて、スロットあたり100万レコード、インデックスあたり1,000万レコード、かつスロットあたり5GBのレコードサイズというハードコーディング制限がありました。しかし本リリースではこの機能はデフォルトで無効になっているため、この機能を有効に設定し、各パラメータに適切なオーバーロード制限（オプション）を設定する必要があります。

表 3: CDLのオーバーロードデータの防止

コマンド	変更内容
古いコマンド： cdl datastore session overload-protection disable true	古いCDLオーバーロード防止コマンドは廃止されます。 (注) コマンドは機能しませんが、後方互換性のためにのみ使用できます。
新しいコマンド： cdl datastore session features overload-protection enable <true/false>	オーバーロード防止の設定は、オーバーロード防止とアラートを設定する cdl データストアセッション機能 の設定の下に移動されます。 CDLでは現在、次のものを設定することが可能です。 - スロット/インデックスあたりのレコードキャパシティ - スロットあたりのレコードキャパシティ (バイト単位) - メジャーおよびクリティカルアラームの設定 (%)

オーバーロード防止が有効になっている場合は、アラートも有効になります。prometheus-rules-cdlポッドが生成されます。オーバーロード防止が無効になっている場合、アラートは無効になります。prometheus-rules-cdlポッドが削除されます。

オーバーロード防止パラメータの設定

オーバーロード防止の制限を設定するために、次のパラメータが設定されています。

- **cdl datastore session features overload-protection index-max-record-count** *<value>*
- **cdl datastore session features overload-protection slot-max-record-count** *<value>*
- **cdl datastore session features overload-protection slot-max-size** *<value>*
- **cdl datastore session features overload-protection hard-limit-percentage** *<value>*

次の表に、設定の詳細を示します。

表 4: オーバーロード防止パラメータの設定

CLI コマンド	説明
cdl datastore session features overload-protection enable <i><true/false></i>	(オプション) CDL オーバーロード防止はデフォルトでは無効になっています。デフォルト値は false です。
cdl datastore session features overload-protection index-max-record-count <i><value></i>	(オプション) インデックスシャードに保存できるレコードの最大数。 デフォルト値は 60000000 (60M) です。 指定できる範囲は 100k ~ 100M です。 (注) 100 ~ 1000 の範囲は、ラボ環境でのテストにのみ適用されます。実稼働環境での適用はお勧めしません。
cdl datastore session features overload-protection slot-max-record-count <i><value></i>	(オプション) スロットシャードに保存できるレコードの最大数。 デフォルト値は 2500000 (2.5M) です。 範囲は 100 または 100k ~ 10M のいずれかです。 (注) 100 の値は、ラボ環境でのテストにのみ適用されます。実稼働環境での適用はお勧めしません。

CLI コマンド	説明
cdl datastore session features overload-protection slot-max-size <i><value></i>	(オプション) スロットシャードの最大サイズ (メガバイト単位)。 デフォルト値は 16384 (16GB) です。 範囲は 1GB ~ 96GB です。
cdl datastore session features overload-protection hard-limit-percentage <i><value></i>	(オプション) ソフト制限に加える、追加のキャパシティ (パーセント)。これは、CDL エンドポイントで更新要求を拒否する場合を判断するために使用されます。たとえば、index shard = 1、index-record-capacity = 100、hard-limit-percentage = 5 の場合、インデックスレコードの数が 100 のときは作成要求が拒否され、更新要求は 105 に達した場合にのみ拒否されます。 デフォルト値は 5 です。指定できる範囲は 0 ~ 10 です。
cdl datastore session features overload-protection major-alert-threshold <i><value></i>	(オプション) CDL がアラート <i>cdlOverloaded-major</i> をトリガーするしきい値 (パーセンテージ)。 デフォルト値は 80 です。 指定できる範囲は 40 ~ 100 です。
cdl datastore session features overload-protection critical-alert-threshold <i><value></i>	(オプション) CDL がアラート <i>cdlOverloaded-critical</i> をトリガーするしきい値 (パーセンテージ)。 デフォルトは 90 です。 指定できる範囲は 40 ~ 100 です。

アラートのパーセンテージを設定するには、次のコマンドを実行します。

```
cdl datastore session features overload-protection critical-alert-threshold <percentage>
```

```
cdl datastore session features overload-protection major-alert-threshold <percentage>
```

アラートの確認

CEE Ops Center CLI で次のコマンドを使用して、アラートを確認できます。

```
show alerts active { detail | summary }
```

また、次のコマンドを使用して、CEE Ops Center CLI でアラートをフィルタ処理できます。

```
show alerts active detail summary "CDL is overloaded."
```

次の例では、システムキャパシティが 80% 以上になった場合に *cdlOverloaded - major* アラートがトリガーされます。

```
alerts active detail cdlOverloaded-major 5446095ab264
severity      major
type          "Processing Error Alarm"
startsAt      2020-10-15T15:09:00.425Z
source        System
summary       "CDL is overloaded."
```

次の例では、システムキャパシティが 90% 以上になった場合に *cdlOverloaded - critical* アラートがトリガーされます。

```
alerts active detail cdlOverloaded-critical 5446095ab264
severity      critical
type          "Processing Error Alarm"
startsAt      2020-10-15T15:09:16.425Z
source        System
summary       "CDL is overloaded."
```

CDL のラック変換

機能説明

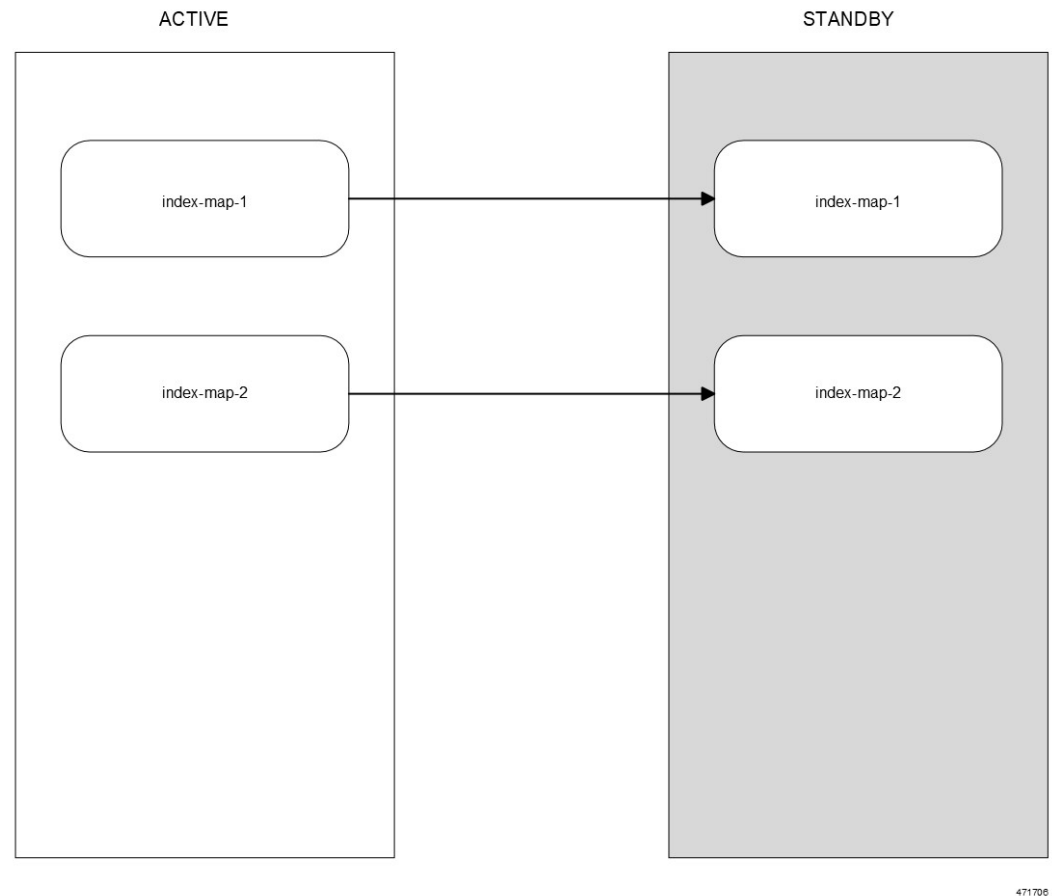
CDL は、再調整メソッドを使用したトラフィックのドレインなしでの、ハーフハイトからフルハイトへのラック変換をサポートしています。

この再調整メソッドは、移行中の既存のコールと新しいコールには影響しません。この機能は、CDL のインデックスマップの再調整に対応します。

CDL 変換の MOP

CDL のハーフラック構成をフルラック構成に移行するには、次の手順を使用します。

1. ラック 1 をアクティブ、ラック 2 をスタンバイに設定します。



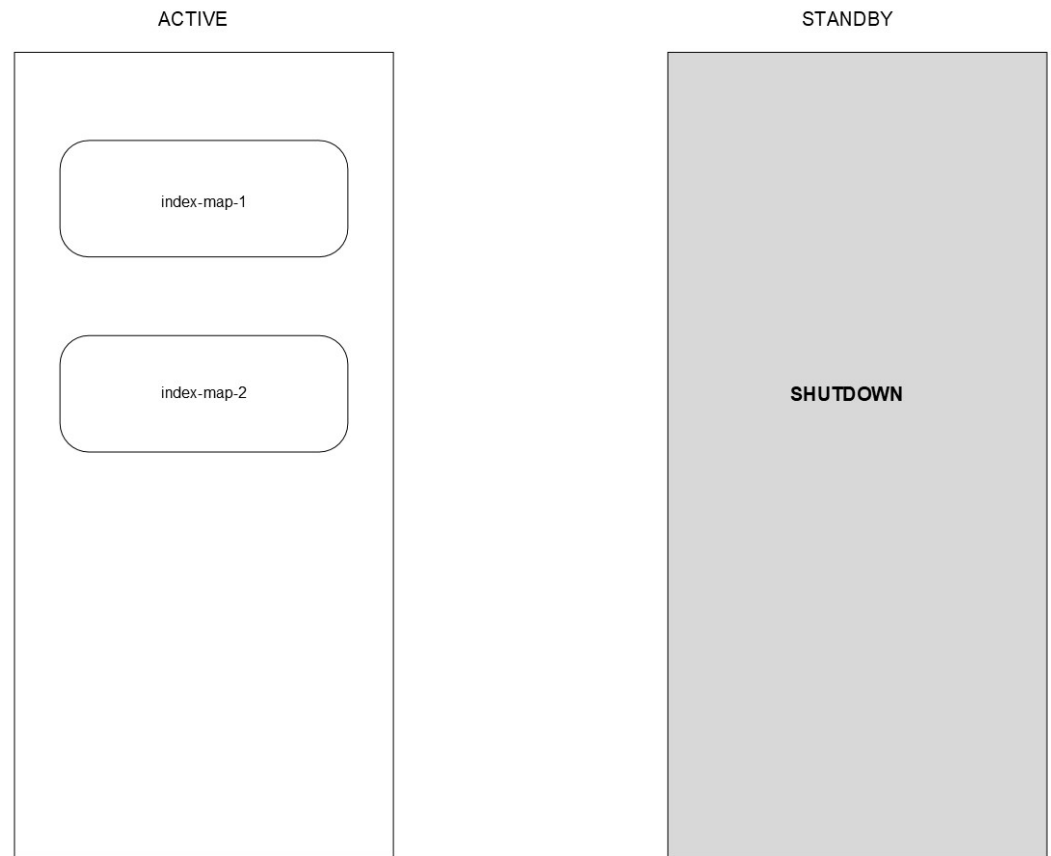
471706

前提条件：CDLはハーフラック構成を使用して実行されていますが、両方のサイトにフルラックのリソースを備える必要があります。

アクション：ラック 1 をアクティブにし、ラック 2 をスタンバイにします。CDL の設定変更はありません。

事後検証：トラフィックはラック 1 にのみ到達します。

2. ラック 2 でシステム モードシャットダウンを設定します。



471707

前提条件：ラック 1 はアクティブモード、ラック 2 はスタンバイモードである必要があります。

アクション：ラック 2 でシステム モード シャットダウンを実行します。

事後検証：サイト 1 の *mirror-maker* は実行状態ではありません。

3. ラック 2 をスタンバイに設定します。新しい設定に合わせてフルラック CDL を構成し、`prev-map-count` が設定されたスケールアップモードを有効にします。

以下に設定例を示します。

configure

```
cdl datastore session mode scale-up
cdl datastore session index prev-map-count 2
```

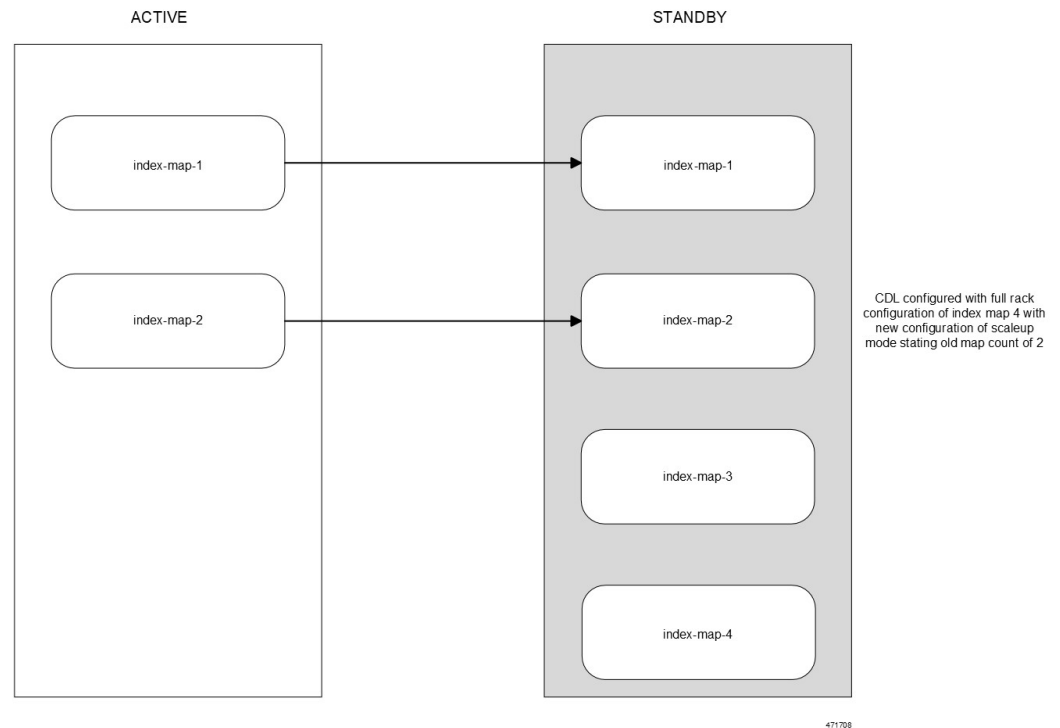
スケールアップモードなしのハーフラック構成：

```
cdl datastore session
label-config session
  index map 2
  slot map 4
exit
```

スケールアップモードを使用するフルラック構成：

```
cdl datastore session
label-config session
mode scale-up
  index map 4
  index prev-map-count 2
  slot map 8
exit
```

4. ラック 2 の system mode running を有効にします。

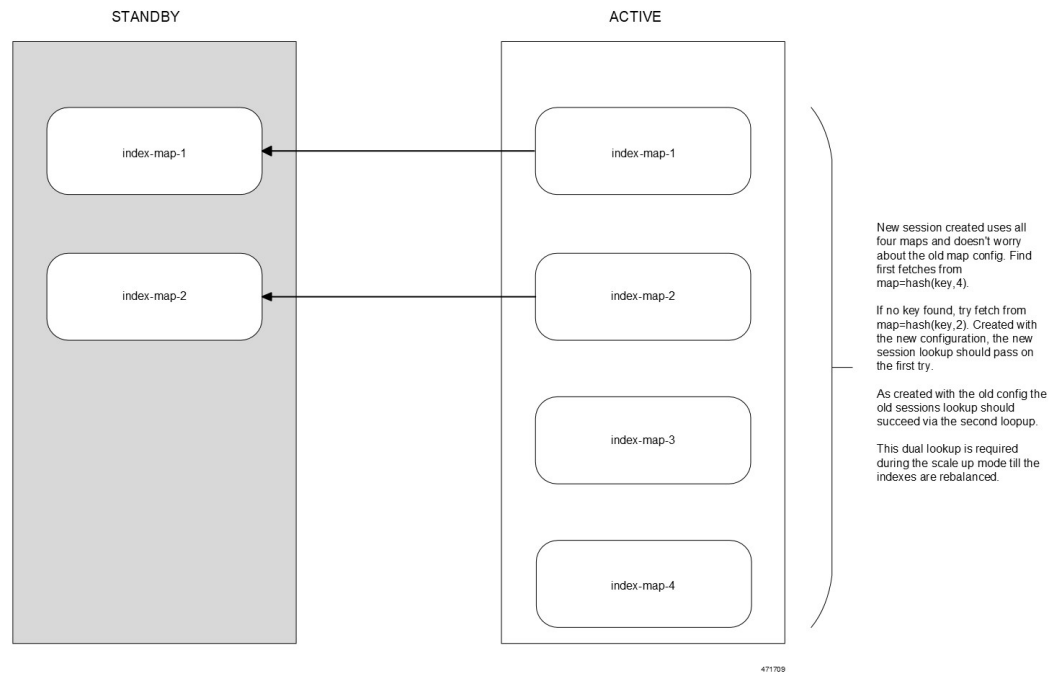


前提条件： ラック 1 はアクティブである必要があり、ラック 2 はフルラック CDL 構成でスタンバイである必要があります。

アクション： ラック 2 で *system mode running* を実行します。

事後検証： 両方のサイトの *mirror-maker* が実行状態である必要があります。サイト 2 の古いインデックスまたはスロットが、リモートピアとの初期同期を実行できる必要があります。

5. トラフィックをラック 2 に切り替えます。

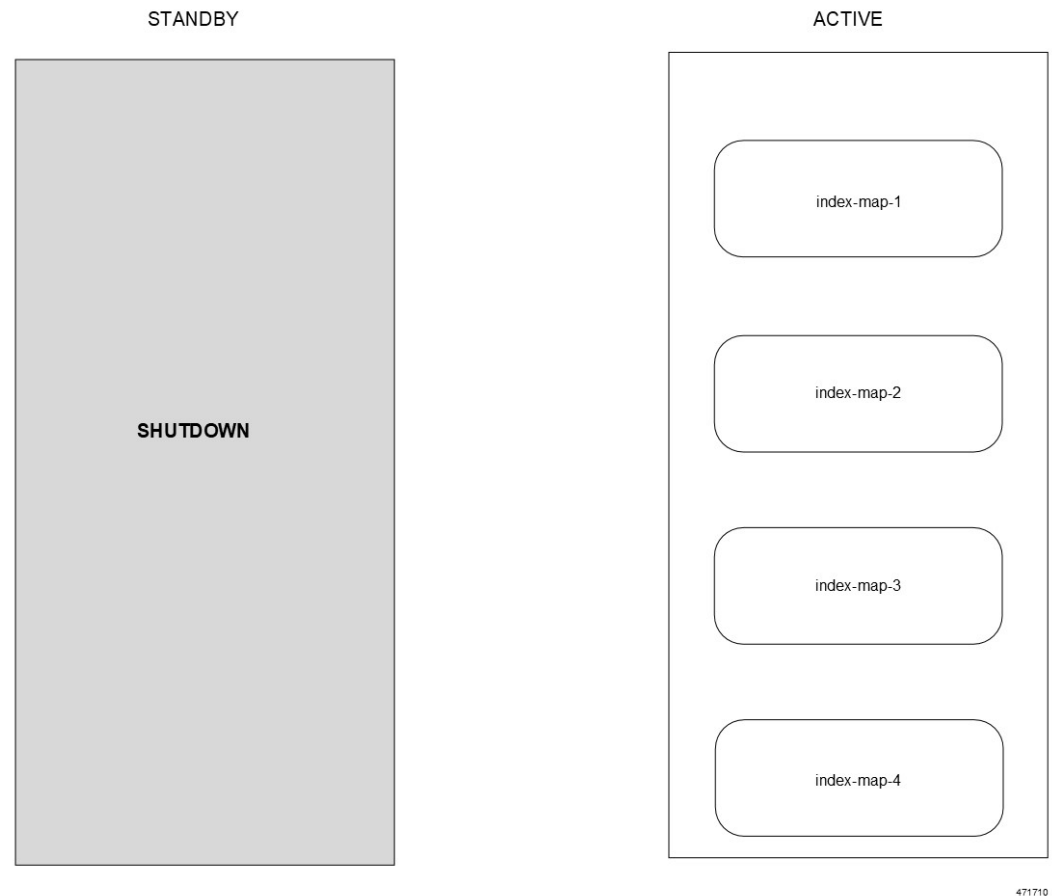


前提条件：ラック 1 はアクティブになっており、ハーフラック構成で実行されている CDL を含むフルラックリソースを備えている必要があります。ラック 2 はスタンバイになっており、フルラック構成で実行されている CDL を含むフルラックリソースを備えている必要があります。

アクション：ラック 1 をスタンバイにし、ラック 2 をアクティブにします。CDL で設定を変更する必要はありません。

事後検証：トラフィックはラック 2 にのみ到達します。

6. ラック 1 のシステム モードシャットダウンを有効にします。



前提条件：ラック 2 はアクティブ、ラック 1 はスタンバイである必要があります。

アクション：ラック 1 でシステム モードシャットダウンを実行します。

事後検証：サイト 2 の *mirror-maker* は実行状態ではありません。

7. ラック 1 をスタンバイに設定します。新しい設定でフルラック CDL を構成し、premap-count が設定されたスケールアップモードを有効にします。

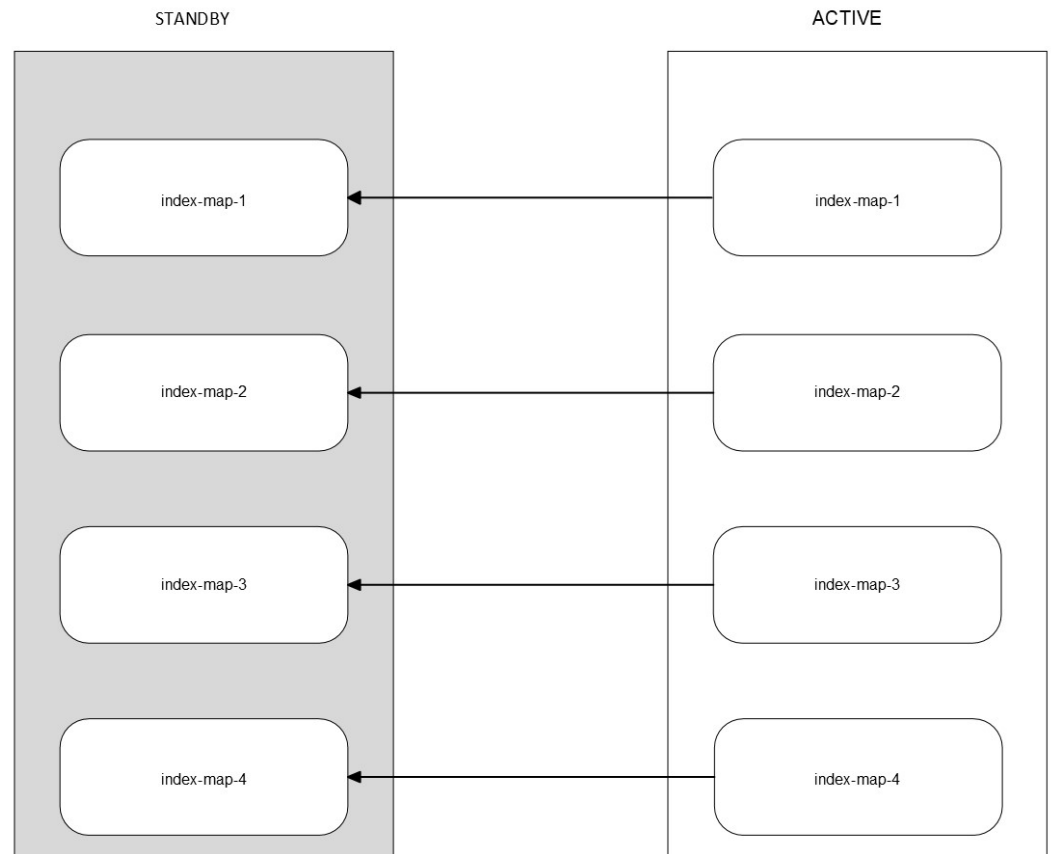
以下に設定例を示します。

configure

```
cdl datastore session mode scale-up
cdl datastore session index prev-map-count 2
```

フル構成により、スロットマップ数とインデックスマップ数が更新されます。新しく追加されたスロットとインデックスのラベル設定が追加されます。

8. ラック 1 の system mode running を有効にします。



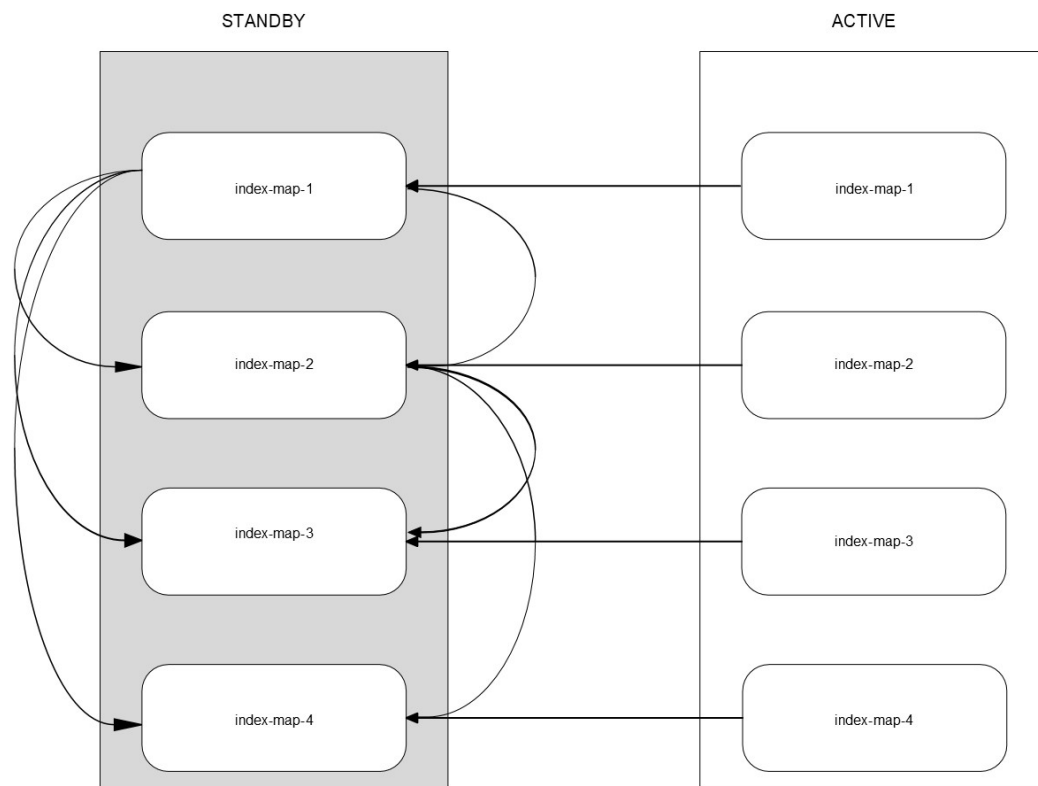
471711

前提条件：ラック 2 はアクティブである必要があります、ラック 1 はフルラック CDL 構成でスタンバイである必要があります。

アクション：ラック 1 で *system mode running* を実行します。

事後検証：両方のサイトの *mirror-maker* が実行状態である必要があります。サイト 1 のすべてのインデックスとスロットが、リモートピアとの初期同期を実行できる必要があります。CDL レプリケーションエラーがあるかどうかを CEE アラートで確認します。

9. ラック 1 (スタンバイ) の CLI を使用してインデックスの再調整をトリガーします。



471712

前提条件：CDL が両方のサイトで正常な状態になっている必要があります。ラック 1 はスタンバイ、ラック 2 はアクティブである必要があります。両方のサイトに、スケールアップモードがインデックス *prev-map-count* で設定されている必要があります。

ラック 1 で CLI によるインデックスの再調整を行うと、（古いマップ数に応じて）以前計算されたマップから（新しいマップ数に応じて）新しく計算されたマップにインデックスがコピーされます。これは、コピーが完了し、以前計算されたマップからインデックスを削除した後に行われます。

アクション：スタンバイサイト（ラック 1）で次のコマンドを使用して、*rebalance-index* の実行をトリガーします。

```
cdl rebalance-index run tps tps_value
```

事後検証：再調整ステータスを追跡するには、次の手順を実行します。

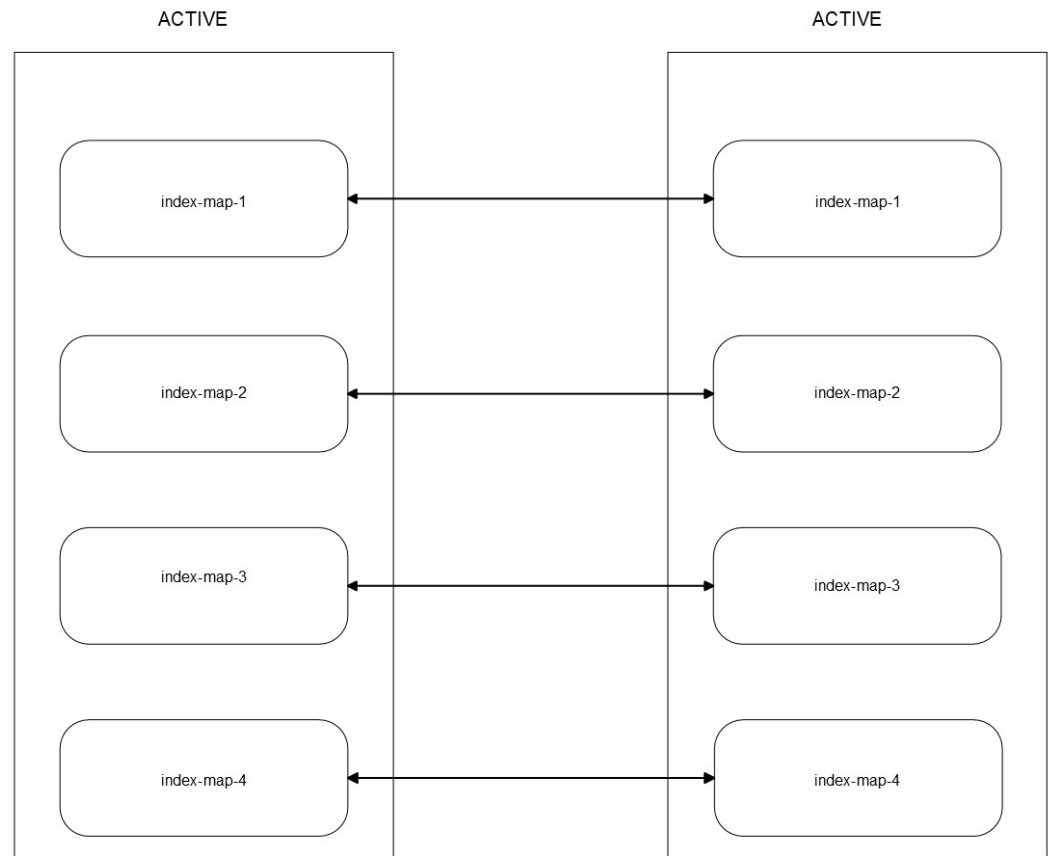
- 次の CLI コマンドを使用して、再調整ステータスをモニターします。

```
cdl rebalance-index status
```

- 次の CLI コマンドを使用して、再調整の実行を検証します。

```
cdl rebalance-index validate
```

10. インスタンスをアクティブ-アクティブに切り替えます。



471713

前提条件：再調整が正常に完了する必要があります。

アクション：ラック 1 とラック 2 をアクティブとして有効にします。

事後検証：トラフィックが両方のサイトに到達する必要があります。

11. 両方のサイトから CDL スケールアップモード設定を削除します。

前提条件：ラック 1 とラック 2 の両方がアクティブである必要があります。

アクション：両方のサイトで次のコマンドを設定します。

no cdl datastore session mode

事後検証：CDL が正常な状態で実行されている必要があります。スケールアップモードが無効になると、*mirror-maker* が再起動します。

トラブルシューティング

再調整ログを表示するには、CLI で次のロガーを設定します。

```
cdl logging logger datastore.idx-rebalance.session
level warn
exit
```

再調整されたインデックスキーの総数を把握するために、インデックスポッドに次のメトリックが導入されています。

メトリック	ラベル
index_rebalanced_keys_total	cdl_slice、shardId

CDL 用の FindAll および FindAllNotify クエリの機能拡張

機能説明

このリリースでは、CDL は FindAll クエリと FindAllNotify クエリの拡張機能をサポートしています。

次の操作をサポートするために、新しいクエリフィルタが導入されています。

- FindAllNotify クエリに最大 20 の IN フィルタ。CDL はスロットからすべてのセッションを取得した後、各セッションを渡されたフィルタと照合します。セッションキーが渡されたフィルタのいずれかと一致した場合、一致した IN フィルタ条件とともに NF に通知します。
- 複数の条件を持つ NOT-MATCH 操作。
- CDL CLI の IN 操作と NOT 操作。
- 通知の IPC ストリーミング。

以前のリリースでは、CDL はアプリケーションへの通知の送信に単項 RPC を使用していました。このリリースでは、CDL は自身とアプリケーション間の IPC を最適化するために、双方向ストリーミング RPC をサポートしています。

機能の仕組み

このセクションでは、FindAll クエリと FindAllNotify クエリでさまざまなフィルタ条件をサポートする、CDL の動作のシーケンスについて説明します。

IN フィルタのサポート

1. システムは、クエリパラメータ内のすべての AND 条件と IN 条件をチェックします。
 - クエリに AND と IN の両方のクエリフィルタが含まれている場合は、次の方法で処理されます。

- CDL は、まずすべての AND フィルタをチェックします。いずれかの AND フィルタが失敗した場合、そのキーはフィルタに一致しなかったため、他の AND フィルタと IN フィルタはスキップされます。
 - すべての AND フィルタが一致したら、IN フィルタをチェックします。いずれかの IN フィルタが一致した場合、そのキーはすでにフィルタに一致しているため、CDL は他の IN フィルタをスキップします。
 - クエリに IN フィルタのみが存在し、いずれかの IN フィルタが一致した場合、CDL は他の IN 条件の処理をスキップします。
2. クエリ要求に 20 を超える IN フィルタがある場合は、400 Bad Request メッセージで応答を返します。
 3. CDL は、FindAllNotify クエリ要求で有効になっている場合にのみ、通知とともに一致した IN フィルタ条件を送信します。



(注) 要求でより多くの IN フィルタが渡されると、クエリのパフォーマンスに影響します。

複数の条件での **NOT-MATCH** 操作のサポート

CDL は、このクエリのフィルタ処理で次の追加条件をサポートしています。

- not-match
- not-starts-with
- not-ends-with
- not-contains



(注) セッションのいずれかのキー (pk/uk/nuk) が **match**、**contains**、**starts-with**、または **ends-with** の条件を満たす場合、そのセッションは **not-match**、**not-contains**、**not-starts-with**、または **not-ends-with** の条件のそれぞれに対応するフィルタには選択されません。

CDL CLI での IN 操作と NOT 操作のサポート

- CLI でセッションをフィルタ処理するために、次の追加の条件がサポートされています。
 - が次の文字列を含む (contains)
 - not-contains
 - not-ends-with
 - not-starts-with

- not-match

- CLI での IN 操作では、AND フィルタ **filter** に加えて、**in-filter** フィルタもサポートされます。

次のコードスニペットは、CLI コマンドの例です。

```
cdl show sessions summary filter { condition ends-with key 6 } in-filter { in-condition starts-with key key-1 } in-filter { in-condition starts-with key key-2 }
```



(注) **filter** および **in-filter** フィルタは、条件のリストです。クエリは最大 20 の **in-filter** フィルタをサポートします。**filter** で言及されているフィルタはグループ化されているため、フィルタ、**filter**、および **in-filter** の順序は重要ではありません。同様に、**in-filter** で言及されているフィルタもグループ化されています。

- **filter** と **in-filter** によるフィルタ処理では、次の CDL 操作がサポートされています。
 - **cdl show sessions count summary**
 - **cdl show sessions count detailed**
 - **cdl show sessions detailed**
 - **cdl show sessions summary**
 - **cdl clear sessions**

ストリーミング通知のサポート

- CDL は、双方向ストリーミング RPC を使用して、トランザクション ID を持つストリームを介して通知をアプリケーションに送信します。応答は、関連付けのためのトランザクション ID とともに、アプリケーションからデータストアに異なるストリームで送信されます。
- デフォルトでは、CDL は単項 RPC を使用し、ストリーミング機能は無効になっています。ストリーミングは、機能フラグ **use-stream** が CDL CLI で設定されている場合にのみ有効になります。ストリーミングが有効になっている場合、CDL 通知はすべてストリーミングを使用して送信されます。

通知のストリーミング機能を有効にするには、次の CLI コマンドを使用します。

```
cdl datastore session slot notification use-stream true
```

- ストリーミングが有効になっている場合、デフォルトでは、CDL から通知エンドポイントへ 4 つのストリーム接続があります。接続数は CDL CLI を使用して設定できます。ストリーム接続はラウンドロビン方式で使用されます。

- スロットまたはエンドポイントから通知エンドポイントへのアクティブなストリーム接続の数を確認するためのメトリックをサポートします。これらのメトリックを使用して、アクティブなストリーム接続がない場合にアラートを生成します。

ネットワーク ポリシー設定

機能の概要と変更履歴

要約データ

該当製品または機能エリア	Cloud Native Broadband Network Gateway (cnBNG) 2022.02 以降
該当するプラットフォーム	ベアメタル、OpenStack、VMware
機能のデフォルト設定	無効：設定が必要
このリリースでの関連する変更点	N/A
関連資料	<i>UCC CDL Configuration and Administration Guide</i> リリース 1.8

マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	CDL 1.8.0

機能説明

このリリースでは、CDL を使用して、ネットワークポリシーを設定しグローバル ネットワーク ポリシー設定をオーバーライドすることができます。設定が有効になっている場合、CDL エンドポイント、インデックス、スロット、Kafka、およびZookeeperのネットワークポリシーが適用されます。

etcd のネットワークポリシーを設定して、グローバル ネットワーク ポリシー設定をオーバーライドすることもできます。設定が有効になっている場合、etcdのネットワークポリシーが適用されます。

CDL ネットワークポリシーの設定

CDL ネットワークポリシーを設定するには、次の CLI コマンドを使用します。

```
config
  cdl network-policy enable <true/false>
```

etcd ネットワークポリシーを設定するには、次の CLI コマンドを使用します。

```
config
  etcd network-policy enable <true/false>
```

トラブルシューティング情報

このセクションでは、CDL 設定に関する既知の問題の一部を解決する方法について説明します。CDL 設定の障害対応に進む前に、次の点を考慮してください。

- **system-id** は、サイト全体で一意です。
- **cluster-id (datastore id)** は、サイト内で一意です。
- CDL の展開後は、インデックスのレプリカを変更できません。レプリカを変更するには、次の手順を実行します。
 1. システムモードを「シャットダウン」に設定します。
 2. レプリカを変更します。
 3. システムモードを「実行」に設定します。
- エンドポイント、インデックス、スロット、および Kafka ポッドのレプリカは、次のラベルと値を持つ k8 ノードの数以下である必要があります。
 - ラベル : `smi.cisco.com/node-type`
 - 値 : `cdl/node-type` の値マルチノード設定の場合、デフォルト値は `session` です。
- Zookeeper および ETCD ポッドのレプリカは、次のラベルと値を持つ k8 ノードの数以下である必要があります。
 - ラベル : `smi.cisco.com/node-type`
 - 値 : `oam`
- Geo レプリケーションのセットアップ（サイトの **cluster-id** 間）では、レプリカと、インデックスおよびスロットポッドのマップを同一にする必要があります。そうしないと、データのレプリケーションが失敗する可能性があります。

CDL インデックスが正確にレプリケートされない

このセクションでは、CDL インデックスポッドを正確にレプリケートする方法について説明します。

問題の説明

セッションデータがリモートサイトに正しくレプリケートされない。

問題の特定

一方のサイトで追加されたデータに、もう一方のサイトではアクセスできません。

考えられる原因

不適切な Geo レプリケーションの設定がこの問題の原因となっている可能性があります。

対処法

この問題を解決する手順は、次のとおりです。

- ローカルの `system-id` とリモートサイトの設定を確認します。
- 各サイト間の CDL エンドポイントと `Kafka` が到達可能かどうかを確認します。
- 各サイトでマップ、インデックスのレプリカ、およびスロットを確認します。それらはすべてのサイトで同一である必要があります。

CDL の操作が失敗しているのに、接続は成功する

このセクションでは、CDL 操作の失敗の問題を解決する方法について説明します。

問題の説明

NF は CDL に接続しているが、検索、作成、更新、削除などのセッション操作は失敗する。

問題の特定

NF ログファイルを確認すると、コールの失敗を特定できます。

考えられる原因

この問題は、CDL インデックスとスロットポッドの準備ができていない場合に発生する可能性があります。

対処法

この問題を解決する手順は、次のとおりです。

- すべてのポッドの準備ができており、実行状態であるかどうかを確認します。
- インデックスポッドは、ピアレプリカ（使用可能な場合はローカルまたはリモート）との同期が完了した場合にのみ、準備完了状態に移行します。
- スロットポッドは、ピアレプリカ（使用可能な場合はローカルまたはリモート）との同期が完了した場合にのみ、準備完了状態に移行します。

- エンドポイントを準備完了状態に移行するには、少なくとも1つのスロットとインデックスポッドが使用可能である必要があります。準備ができていない場合でも、クライアントは GRPC 接続を受け入れます。

CDL のポッドがダウンしている

このセクションでは、CDL ポッドがダウンしている場合にこれらを起動する方法について説明します。

問題の説明

CDL の設定が正しくないため、CDL ポッドが「実行」状態にならない。

問題の特定

次のコマンドを使用して、「describe pods」の出力（コンテナ、メンバー、状態、理由、またはイベント）を確認し、ポッドがダウンしているかどうかを識別します。

```
kubect1 describe pods -n <namespace> <failed pod name>
```

考えられる原因

考えられる原因は次のとおりです。

- ポッドが「*pending*」の状態である。
- ポッドが「*CrashLoopBackOff*」障害の状態である。
- ポッドが「*ImagePullBack*」障害の状態である。

対処法

- ポッドが「*pending*」状態の場合：
 - ラベル値 *cdl/node-type* を持つ k8s ノードが存在するかどうかを確認します。また、レプリカの数、ラベル値 *cdl/node-type* を持つ k8s ノードの数以下であることも確認します。

```
kubect1 get nodes -l smi.cisco.com/node-type=<value of cdl/node-type, default value is 'session' in multi node setup>
```

- ポッドが「*CrashLoopBackOff*」障害の状態の場合：

- ETCD ポッドのステータスを確認します。

```
kubect1 describe pods -n <namespace> <etcd pod name>
```

ETCD ポッドが実行されていない場合は、ETCD の問題を解決してポッドを起動します。

- ポッドが「*ImagePullBack*」障害の状態の場合：

- helm リポジトリとイメージレジストリにアクセスできるかどうかを確認します。
- 必要なプロキシサーバーと DNS サーバーが設定されているかどうかを確認します。

準備中の状態の Mirror Maker ポッド

このセクションでは、「準備中 (*Not Ready*)」の状態のままの Mirror Maker ポッドを移動する方法について説明します。

問題の説明

リモートサイトでの接続の問題により、Mirror Maker ポッドが実行状態ではない。

問題の特定

次のコマンドを使用して、Describe ポッドの出力およびポッドログを確認します。

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

考えられる原因

リモートサイトの Kafka ブローカとの接続の問題がこの問題の原因となっている可能性があります。

対処法

この問題を解決する手順は、次のとおりです。

- Kafka 用に設定された外部 IP が正確であるかどうかを確認します。
- 外部 IP を介して Kafka のリモートサイトに到達できるかどうかを確認します。

CDL からのレコード消去の通知が早い、または遅延する

このセクションでは、CDL からのレコード消去について送信される通知が早い、または遅延する問題を解決する方法について説明します。

問題の説明

CDL からのレコード消去について送信される通知が早い、または遅延する。

問題の特定

CDL からのレコードの消去中に来る通知が早い、または遅延します。

考えられる原因

Kubernetes クラスタ内のノードの時刻が同期されていない。

対処法

この問題を解決する手順は、次のとおりです。

- k8s クラスタ内のすべてのノードの時刻が同期されているかどうかを確認する。
- すべての k8s ノードの同期ステータスを確認する。

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

CDL でセッションが蓄積する、またはオーバーロードする

このセクションでは、CDL でのセッションの蓄積を解決する方法について説明します。

問題の説明

ネットワークの問題または CDL 上の想定外のセッション数によりセッション数が蓄積し、ピアセッション数と一致しない。

問題の特定

CLI または Grafana ダッシュボードからのセッション数に、予想よりも多くのセッション数が表示されているか、セッション数が増え続けています。

考えられる原因

CDL で受信した *Create* 要求が *Delete* 要求よりも多くなっています。結果として、セッション数は増加し続けます。

対処法

この問題を解決するには、各 NF がサブスクリバのクリアをトリガーするための適切なアクションを実行する必要があります。

ユーザー定義のスケジュール設定とポッドの配置

このセクションでは、CDL ポッドでのスケジュール設定、ノードアフィニティ、およびポッド配置に単一のノードラベルを使用することによって発生する可能性のある、スケジュール不可のポッドの問題を解決する方法について説明します。

問題の説明

nodeType パラメータは、CDL でのスケジュール設定、ノードアフィニティ、およびポッド配置を制御する。現在、このパラメータは単一のノードラベル値 *smi.cisco.com/node-type* で設定されている。CDL がフルノードキャパシティに設定されている場合、単一のノードラベルを使用することによって、（場合により）ポッドがスケジュール不可とマークされる。

問題の特定

CDL ポッドがスケジュール不可とマークされます。

考えられる原因

スケジュール設定、ノードアフィニティ、およびポッド配置に単一のノードラベルを使用している。

対処法

個別のノードラベルを使用して、さまざまな CDL ポッドタイプを制御できます。個別のノードラベルを作成するには、次の手順を実行します。

1. (必須) SMI Cluster Manager Ops Center CLI を介して CDL に使用するすべてのノードで `smi.cisco.com/node-type` ラベルを作成します。
2. 次の表で指定されているように、CDL ポッドのラベルを作成します。

ポッド	ラベル	値
CDL エンドポイントポッド	<code>smi.cisco.com/cdl-ep</code>	<code>true</code>
CDL スロットポッド	<code>smi.cisco.com/cdl-slot-<i><MAP Id 1..n></i></code>	<code>true</code>
CDL インデックスポッド	<code>smi.cisco.com/cdl-index-<i><MAP Id 1..n></i></code>	<code>true</code>



(注) この設定は、ユーザー定義のスケジュール設定にのみ必要です。

例

次の例は、2つのエンドポイントレプリカ、4つのスロットマップ（それぞれ2つのレプリカ）、および2つのインデックスマップ（それぞれ2つのレプリカ）を持つ4つの異なるノードを使用する CDL に個別のノードラベルを作成する方法を示しています。

ノード 1	ノード 2	ノード 3	ノード 4
ポッド : slot-1-0	ポッド : slot-1-1	ポッド : slot-2-0	ポッド : slot-2-1
ラベル : cdl-slot-1	ラベル : cdl-slot-1	ラベル : cdl-slot-2	ラベル : cdl-slot-2
ポッド : slot-3-0	ポッド : slot-3-1	ポッド : slot-4-0	ポッド : slot-4-1
ラベル : cdl-slot-3	ラベル : cdl-slot-3	ラベル : cdl-slot-4	ラベル : cdl-slot-4
ポッド : index-1-0	ポッド : index-1-1	ポッド : index-2-0	ポッド : index-2-1

ノード 1	ノード 2	ノード 3	ノード 4
ラベル : cdl-index-1	ラベル : cdl-index-1	ラベル : cdl-index-2	ラベル : cdl-index-2
ポッド : ep-0			ポッド : ep-2
ラベル : cdl-ep	ラベル : cdl-ep	ラベル : cdl-ep	ラベル : cdl-ep

設定例 : SMI Cluster Manager

```
clusters user-k8s
  nodes session1
    k8s node-labels smi.cisco.com/cdl-ep true
    exit
    k8s node-labels smi.cisco.com/cdl-index-1 true
    exit
    k8s node-labels smi.cisco.com/cdl-slot-1 true
    exit
    k8s node-labels smi.cisco.com/cdl-slot-3 true
    exit
    k8s node-labels smi.cisco.com/node-type session
    exit
  exit
exit
clusters user-k8s
  nodes session2
    k8s node-labels smi.cisco.com/cdl-ep true
    exit
    k8s node-labels smi.cisco.com/cdl-index-1 true
    exit
    k8s node-labels smi.cisco.com/cdl-slot-1 true
    exit
    k8s node-labels smi.cisco.com/cdl-slot-3 true
    exit
    k8s node-labels smi.cisco.com/node-type session
    exit
  exit
exit
clusters user-k8s
  nodes session3
    k8s node-labels smi.cisco.com/cdl-ep true
    exit
    k8s node-labels smi.cisco.com/cdl-index-2 true
    exit
    k8s node-labels smi.cisco.com/cdl-slot-2 true
    exit
    k8s node-labels smi.cisco.com/cdl-slot-4 true
    exit
    k8s node-labels smi.cisco.com/node-type session
    exit
  exit
exit
clusters user-k8s
  nodes session3
    k8s node-labels smi.cisco.com/cdl-ep true
    exit
    k8s node-labels smi.cisco.com/cdl-index-2 true
    exit
    k8s node-labels smi.cisco.com/cdl-slot-2 true
    exit
    k8s node-labels smi.cisco.com/cdl-slot-4 true
    exit
  exit
```

```

    k8s node-labels smi.cisco.com/node-type session
  exit
exit
exit

```

サンプル設定 : CDL

```

cdl label-config session
  endpoint key smi.cisco.com/cdl-ep
  endpoint value true
  slot map 1
    key smi.cisco.com/cdl-slot-1
    value true
  slot map 2
    key smi.cisco.com/cdl-slot-2
    value true
  slot map 3
    key smi.cisco.com/cdl-slot-3
    value true
  slot map 4
    key smi.cisco.com/cdl-slot-4
    value true
  index map 1
    key smi.cisco.com/cdl-index-1
    value true
  index map 2
    key smi.cisco.com/cdl-index-2
    value true
cdl datastore session
  label-config session
  endpoint replica 2
  index replica 2
  index map 2
  slot replica 2
  slot map 4
exit

```

モニタリング

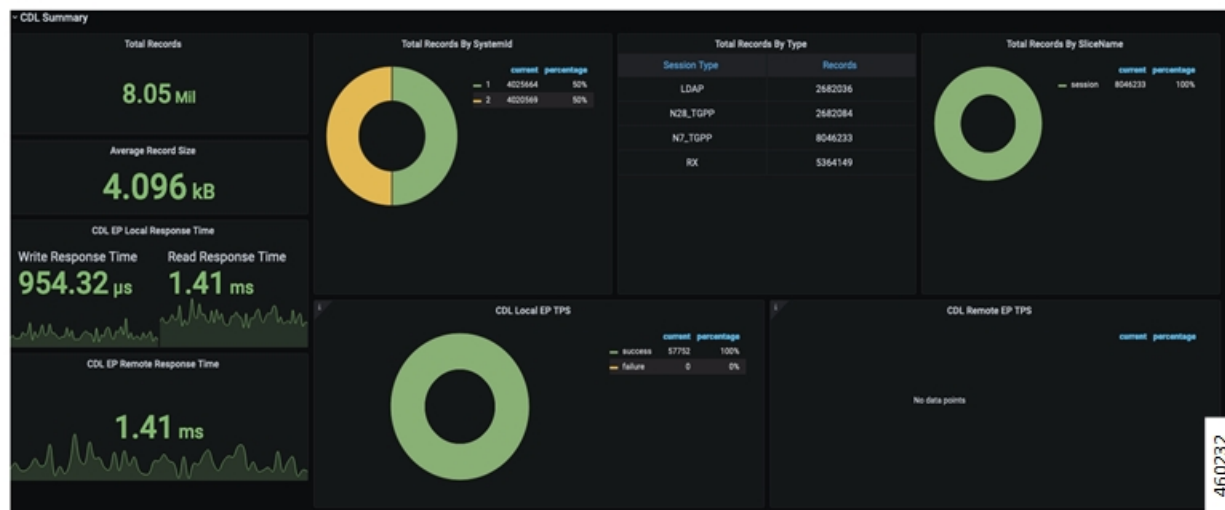
このセクションでは、CDLのステータスとレコードをモニターする方法について説明します。Grafana ダッシュボードを介して CDL をモニターすることもできます。

Grafana ダッシュボードを介した CDL のモニタリング

デフォルトでバンドルされている CDL ダッシュボードと呼ばれる Grafana ダッシュボードを使用して、CDL のさまざまなアクティビティをモニターできます。CDL ダッシュボードには、*cdl-endpoint*、スロット、インデックスなどのさまざまなポッドの CDL に対する TPS と、各操作にかかる応答時間が表示されます。

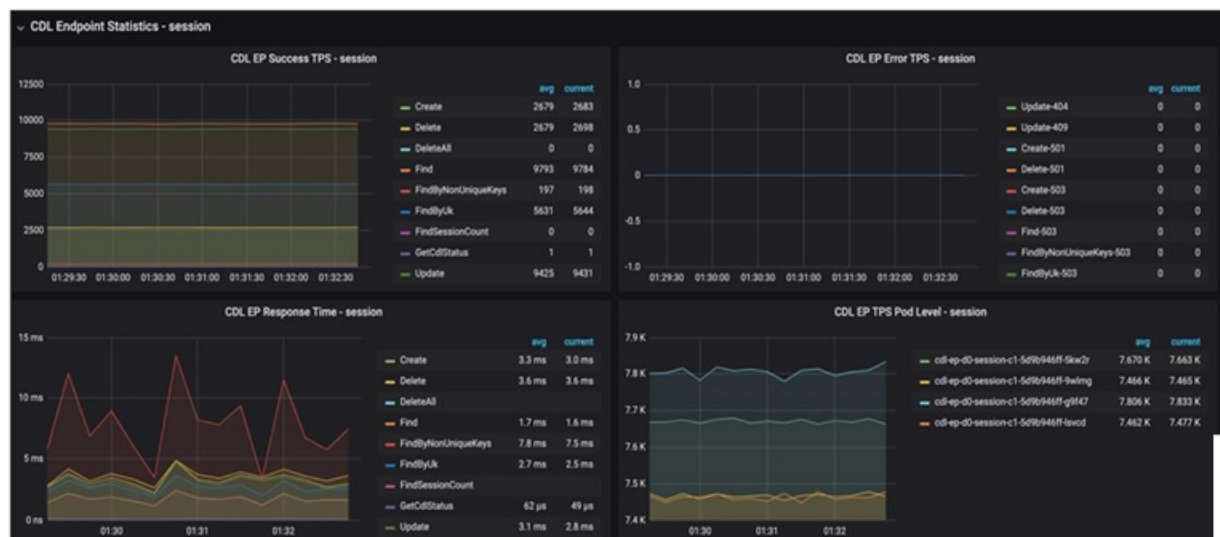
Type、SliceName、SystemID などでも分類した合計レコード数を表示する Grafana CDL Summary ダッシュボードの例を以下に示します。

図 20 : Grafana CDL ダッシュボード : 概要



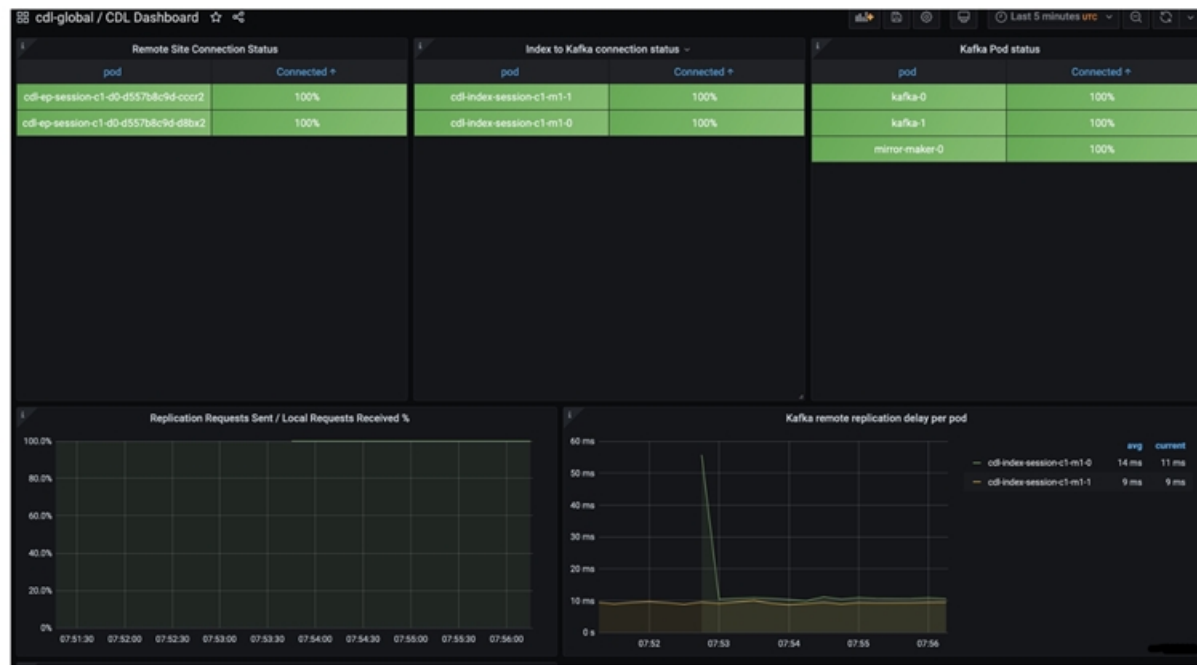
CDL TPS と応答時間のグラフを表示する Grafana CDL ダッシュボードの例を以下に示します。

図 21 : Grafana CDL ダッシュボード : CDL TPS と応答時間



Grafana CDL ダッシュボードの拡張により、Geo レプリケーションのステータスが表示されます。Geo レプリケーションのステータスとその他の詳細を表示する Grafana ダッシュボードの例を以下に示します。

図 22: Grafana CDL ダッシュボード: GR 接続ステータス



GR 接続、インデックス レプリケーションとスロットレプリケーションのパネル、およびそれらの説明を次の表に示します。

表 5: GR 接続ステータス

パネル	説明
リモートサイトの接続ステータス	エンドポイントポッドごとのアクティブなリモートサイト接続（パーセンテージ）。0の状態が5分を超えると、アラートがトリガーされます。
インデックスから Kafka への接続ステータス	各インデックスポッドからの現在アクティブな Kafka 接続の平均値。
Kafka ポッドのステータス	Kafka ポッドと mirrorMaker ポッドの準備ステータス。
送信したレプリケーション要求/受信したローカル要求 %	NFが受信したローカル要求に対する、リモートサイトに送信したレプリケーション要求の比率。この比率が5分を超えて90%を下回ると、アラートがトリガーされます。

パネル	説明
ポッドごとの Kafka リモートレプリケーションの遅延	Kafka を介してパートナーサイトにレコードをレプリケートする際の合計遅延。レコードのレプリケートで 10 秒を超える遅延が 5 分を超えて続くと、アラートがトリガーされます。
ドロップされたリモート要求の総数	キューがいっぱいであるためにドロップされたリモート要求の総数。

表 6: インデックスのレプリケーション

パネル	説明
ポッドごとの Kafka パブリッシュ TPS	Kafka 書き込み要求（パブリッシュ）のインデックスポッドごとの合計レート。
ポッドごとの Kafka リモートレプリケーション TPS	パートナーサイトから受信する Kafka 要求（消費）のインデックスポッドごとの合計レート。

表 7: スロットのレプリケーション

パネル	説明
送信したスロット Geo レプリケーション要求	操作ごとの、cdl-ep からリモートサイトへのレプリケーション要求の合計レート。
受信したスロット Geo レプリケーション要求	操作タイプごとの、スロットポッドが受信したレプリケーション要求の合計レート。
スロットチェックサムの一貫性	スロットチェックサム不一致の合計レート。
スロットの調整	スロットデータのチェックサムの不一致によるスロット調整の合計レート。

CDL のステータスとレコードの確認

CDL Ops Center で **cdl show status** コマンドを使用して、CDL のステータスを確認し、クライアントによって挿入されたデータを調べることができます。**cdl show status** コマンドの詳細については、『*UCC CDL Command Reference Guide*』を参照してください。

次の例は、すべての CDL コンポーネントのステータスとクライアントのデータを示しています。

```
[unknown] cdl# cdl show status
message params: {cmd:status mode:cli dbName:session sessionIn:{mapId:0 limit:10000
filters:[]}}
site {
  system-id 1
```

```
cluster-id 1
db-endpoint {
    state STARTED
}
slot {
    map-id 1
    instance-id 2
    records 100
    capacity 1000000
    state STARTED
}
slot {
    map-id 1
    instance-id 1
    records 100
    capacity 1000000
    state STARTED
}
index {
    map-id 1
    instance-id 2
    records 500
    capacity 2500000
    state ONLINE
}
index {
    map-id 1
    instance-id 1
    records 500
    capacity 2500000
    state ONLINE
}
}
[unknown] cdl#
```




第 2 章

CDL のバックアップと復元

この章では、CDL のバックアップと復元の操作手順について説明します。

- [CDL Ops-Center の設定へのアクセス](#) (77 ページ)
- [バックアップと復元](#) (78 ページ)

CDL Ops-Center の設定へのアクセス

次の手順を使用して、CDL API 設定にアクセスします。

始める前に

CDL API を設定するために、CDL データストアが設定されていることを確認します。

手順

ステップ 1 `cdl api deploy true` コマンドを使用して、CDL API ポッドの展開を有効にします。

デフォルトでは、ポッドの展開は無効になっています。

ステップ 2 次のコマンドを使用して、CDL API ポッドを垂直方向にスケーリングします。

a) `cdl api cpu-limit cpu_limit` コマンドを使用して、最大 CPU 制限を設定します。

例 :

```
cdl api cpu-limit 2000
```

cpu_limit は 1000 ~ 8000 の範囲の整数である必要があります。デフォルト : 4000。

b) `cdl api go-max-procs cpu_cores` コマンドを使用して、CPU コアの最大数を設定します。

例 :

```
cdl api go-max-procs 4
```

cpu_cores は 1 ~ 32 の範囲の整数である必要があります。デフォルト : 4。

- c) **cdl api memory-limit** *memory_limit* コマンドを使用して、許可される最大メモリを Gi 単位で設定します。

例：

```
cdl api memory-limit 6
```

memory_limit は 1 ～ 8 の範囲の整数である必要があります。デフォルト：4。

- d) **cdl api storage** *storage_size* コマンドを使用して、バックアップストレージを GB 単位で設定します。

例：

```
cdl api storage 90
```

storage_size は 1 ～ 200 の範囲の整数である必要があります。デフォルト：90。

バックアップと復元

バックアッププロセスをトリガーして復元するには、次の手順を実行します。

始める前に

cdl actions backup および **cdl actions restore** コマンドにアクセスするように **cdl api deploy true** を設定していることを確認します。

手順

ステップ 1 **cdl actions backup start** [**concurrency-factor** *concurrency_factor* | **created-before** *timestamp* | **slice-name** *slice_name*] コマンドを使用して、バックアッププロセスをトリガーします。

このコマンドはバックアッププロセスをトリガーしますが、プロセスが完了するまで待機はしません。作成されたダンプファイル名とともに成功または失敗のメッセージを返します。バックアッププロセスが完了すると、復元プロセスのために出力のダンプファイルを収集または参照できます。このファイルは、**cdl-api-0** ポッドを実行しているノードの `/data/<namespace>/data-cdl-api-0/<file-name.dump>` に配置できます。

このコマンドを実行すると、これにより **CDL** バックアッププロセスがトリガーされます。よろしいですか? という警告が表示されます。**yes** と入力して、**CDL** バックアッププロセスのトリガーを続行します。

次のパラメータは、**backup start** コマンドで設定されるオプションです。

- **concurrency-factor** *concurrency_factor* : スロットから同時にデータを取得することによってバックアップを取ります。デフォルトでは、一度にクエリされるスロットは 1 つだけです。

concurrency_factor は、1 ～ 8 の範囲の整数である必要があります。

- **created-before** *timestamp* : このタイムスタンプより前に作成されたレコードのバックアップを取ります。タイムスタンプは `CCYY-MM-DDTHH:MM:SS` 形式にする必要があります。

デフォルトでは、使用可能なすべてのデータがバックアップのためにクエリされます。

- **slice-name** *slice_name* : 指定したスライス名のバックアップを取ります。ここでは、設定したスライス名のみ渡すことができます。

デフォルトでは、すべてのスライス名のバックアップを取ります。

ステップ 2 **cdl actions backup status** コマンドを実行してバックアッププロセスのステータスを確認します。

このコマンドは、次のいずれかのステータスを返します。

- **idle | completed** : CDL バックアッププロセスが完了した、または進行中ではない
- **in-progress** : CDL バックアッププロセスが進行中である

ステップ 3 **cdl actions restore start file filename** コマンドを使用して、復元プロセスをトリガーします。

このコマンドは復元プロセスをトリガーしますが、プロセスが完了するまで待機はしません。成功または失敗のメッセージが返されます。このコマンドでは、**file** パラメータの設定が必須です。

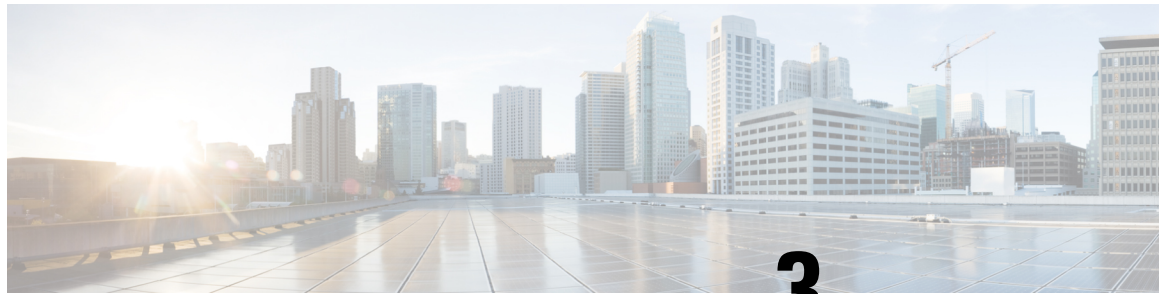
このコマンドを実行すると、これにより **CDL** 復元プロセスがトリガーされ、既存のデータが上書きされます。よろしいですか？という警告が表示されます。**yes** と入力して、CDL 復元プロセスのトリガーを続行します。

ファイル名には **.dump** 拡張子が付いている必要があります。ファイルは、**cdl-api-0** ポッドを実行しているノードの、ホストパス **/data/<namespace>/data-cdl-api-0/<file-name.dump>** に存在する必要があります。

ステップ 4 **cdl actions restore status** コマンドを実行して復元プロセスのステータスを確認します。

このコマンドは、次のいずれかのステータスを返します。

- **idle | completed** : CDL 復元プロセスが完了した、または進行中ではない
- **in-progress** : CDL 復元プロセスが進行中である



第 3 章

CDL のメトリック

- [概要 \(81 ページ\)](#)
- [CDL カテゴリ \(82 ページ\)](#)

概要

この章では、CDL のパフォーマンスのモニターと分析に使用できる重要業績評価指標 (KPI) について説明します。

CDL で使用するメトリックのラベル名と説明は、次の表で定義されています。

表 8: メトリックラベルの説明

メトリックラベル名	ラベルの説明
<i>db</i>	データストアの名前。
動作	CDL ポッドで実行された操作の名前。
<i>errorCode</i>	応答で送信されるエラーコード。
<i>errorMessage</i>	応答で送信されるエラーメッセージ。
<i>slot_shard_id</i>	操作が実行されるスロットマップまたはシャード ID。
<i>slot_instance_id</i>	操作が実行されるスロットインスタンス ID。
<i>shardId</i>	メトリックが固定されるスロットまたはインデックスマップ、またはシャード ID。
<i>instanceId</i>	メトリックが固定されるスロットまたはインデックスマップ、またはインスタンス ID。
<i>session_type</i>	レコードに存在するセッションデータのタイプ。
<i>bucket</i>	バケットは、セッションが存在するバケットを表します。現在のバケットは、1kb 以下、2kb 以下、4kb 以下、8kb 以下、16kb 以下、32kb 超です。

メトリックラベル名	ラベルの説明
<i>notification_type</i>	CDL から送信される通知のタイプ。値：TIMER_EXPIRED、RECORD_CONFLICT、BULK_TASK_NOTIFICATION
<i>topic</i>	Kafka へのパブリッシュ中のトピック
<i>not_found_in</i>	データが検出されなかったポッド。値：インデックス/スロット

CDL カテゴリ

bulk_task_ongoing

説明：任意の時点で処理されている一括タスクの数を示すゲージメトリック

サンプルクエリ：bulk_task_ongoing

ラベル：

- ラベル：db

ラベルの説明：DB 名

例：session

- ラベル：slot_shard_id

ラベルの説明：スロットシャード ID

例：1、2

- ラベル：slot_instance_id

ラベルの説明：スロットインスタンス ID

例：1、2

- ラベル：cdl_slice

ラベルの説明：論理 CDL スライスの名前

例：session

bulk_task_total

説明：処理ステータスを持つ一括タスクの総数

サンプルクエリ：bulk_task_total

ラベル：

- ラベル：db

ラベルの説明：DB 名

例 : session

- ラベル : slot_shard_id

ラベルの説明 : スロットシャード ID

例 : 1、2

- ラベル : slot_instance_id

ラベルの説明 : スロットインスタンス ID

例 : 1、2

- ラベル : cdl_slice

ラベルの説明 : 論理 CDL スライスの名前

例 : session

- ラベル : status

ラベルの説明 : 一括タスクの処理ステータス

例 : timeout、skipped、aborted、completed_last_record、completed

cdl_ep_to_slot_request_tps

説明 : エンドポイントからスロットへの要求 TPS 測定の記録ルール

サンプルクエリ : cdl_ep_to_slot_request_tps

ラベル :

- ラベル : namespace

ラベルの説明 : メトリックの生成元となる Kubernetes 名前空間

例 : cdl-global

ラベル :

- ラベル : pod

ラベルの説明 : メトリックの生成元となるエンドポイントポッド名

- ラベル : operation

ラベルの説明 : DB 操作のタイプ

例 : Create、Update、Delete、UpdateFlags

- ラベル : errorCode

ラベルの説明 : 削除のための DB 応答の errorCode

例 : 0、502

cdl_ep_to_slot_response_time

説明：エンドポイントからスロットへの応答時間測定の記録ルール

サンプルクエリ：cdl_ep_to_slot_response_time

ラベル：

- ラベル：namespace

ラベルの説明：メトリックの生成元となる Kubernetes 名前空間

例：cdl-global

ラベル：

- ラベル：operation

ラベルの説明：DB 操作のタイプ

例：Create、Update、Delete、UpdateFlags

ラベル：

- ラベル：errorCode

ラベルの説明：削除のための DB 応答の errorCode

例：0、502

cdl_geo_replication_enabled

説明：Geo レプリケーションのステータスを示すゲージメトリック。Geo レプリケーションが有効な場合の値は 1 で、そうでない場合は 0 です

サンプルクエリ：cdl_geo_replication_enabled

cdl_index_record_capacity

説明：CDL の合計インデックス レコード キャパシティ

サンプルクエリ：cdl_index_record_capacity{db=\"session\"}

ラベル：

- ラベル：db

ラベルの説明：DB 名

例：session

cdl_slice_state

説明：GR インスタンス認識の CDL スライスのアクティブ状態情報。値が 1 の場合、スライス はアクティブです。

サンプルクエリ：cdl_slice_state

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライスの名前

例 : `session`

ラベル :

- ラベル : `shardId`

ラベルの説明 : スロットシャード ID

例 : 1、2

`cdl_slot_record_capacity`

説明 : CDL の合計スロット レコード キャパシティ

サンプルクエリ : `cdl_slot_record_capacity{db=\"session\"}`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

`cdl_slot_size_capacity`

説明 : CDL の合計スロットサイズキャパシティ

サンプルクエリ : `cdl_slot_size_capacity{db=\"session\"}`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

`consumer_kafka_nonprocessed_records_total`

説明 : 同じポッドから発信されて以降未処理の Kafka レコードの総数

サンプルクエリ : `sum(consumer_kafka_nonprocessed_records_total)by(shardId,instanceId)`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : operation

ラベルの説明 : DB 操作のタイプ

例 : Get、Multi

ラベル :

- ラベル : shardId

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : instanceId

ラベルの説明 : インスタンス ID

例 : 1、2

ラベル :

- ラベル : reason

ラベルの説明 : 消費された Kafka レコードをスキップした理由

例 : old_timestamp

consumer_kafka_records_duration_seconds

説明 : 消費された Kafka レコードの処理に要した時間

サンプルクエリ :

```
sum(irate(consumer_kafka_records_duration_seconds[5m]))by(shardId,instance_id)
```

ラベル :

- ラベル : db

ラベルの説明 : DB 名

例 : session

ラベル :

- ラベル : shardId

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : origin_instance_id

ラベルの説明 : kafka 要求の発信元であるインデックスインスタンス ID。

例 : 1.1、1.2

ラベル :

- ラベル : `systemId`

ラベルの説明 : システムの ID

例 : 1、2

consumer_kafka_records_total

説明 : kafka から消費されたレコードの総数

サンプルクエリ : `sum(irate(consumer_kafka_records_total[5m]))by(shardId,instance_id)`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : `shardId`

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : `origin_instance_id`

ラベルの説明 : kafka 要求の発信元であるインデックスインスタンス ID。

例 : 1.1、1.2

ラベル :

- ラベル : `systemId`

ラベルの説明 : システムの ID

例 : 1、2

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライスの名前

例 : `session`

datastore_internal_requests_duration_seconds

説明：内部データストア要求の処理に要した時間

サンプルクエリ：`sum(datastore_internal_requests_duration_seconds)by(operation)`

ラベル：

- ラベル：db

ラベルの説明：DB 名

例：session

ラベル：

- ラベル：operation

ラベルの説明：DB 操作のタイプ

例：RemoteBulkRead、RemoteBulkReadIndexing、GetChecksumRemoteSlot

ラベル：

- ラベル：errorCode

ラベルの説明：DB 応答の errorCode

例：0、1401

ラベル：

- ラベル：cdl_slice

ラベルの説明：論理 CDL スライスの名前

例：session

datastore_requests_duration_seconds

説明：cdl-ep での要求の処理に要した合計時間

サンプルクエリ：

```
sum(irate(datastore_requests_duration_seconds{errorCode=\"0\",local_request=\"1\"}[5m]))
by (operation)
```

ラベル：

- ラベル：db

ラベルの説明：DB 名

例：session

ラベル：

- ラベル：operation

ラベルの説明 : DB 操作のタイプ

例 : Create、Update、Delete、Find、FindByUK、GetCdlStatus、UpdateFlags

ラベル :

- ラベル : errorCode

ラベルの説明 : DB 応答の errorCode

例 : 0、400、403、404、409、413、501、502、503、507、508

ラベル :

- ラベル : local_request

ラベルの説明 : DB 要求がローカルまたは GR のいずれであるか。local_request = 1 の場合はローカルで、それ以外の場合は GR です。

例 : 1、0

ラベル :

- ラベル : cdl_slice

ラベルの説明 : 論理 CDL スライスの名前

例 : session

datastore_requests_total

説明 : cdl-ep で受信した要求の総数

サンプルクエリ :

```
sum(irate(datastore_requests_total{errorCode=\"0\",local_request=\"1\"}[5m])) by (operation)
```

ラベル :

- ラベル : db

ラベルの説明 : DB 名

例 : session

ラベル :

- ラベル : operation

ラベルの説明 : DB 操作のタイプ

例 : Create、Update、Delete、Find、FindByUK、GetCdlStatus、UpdateFlags

ラベル :

- ラベル : errorCode

ラベルの説明 : DB 応答の `errorCode`

例 : 0、400、403、404、409、413、501、502、503、507、508

ラベル :

- ラベル : `local_request`

ラベルの説明 : DB 要求がローカルまたは GR のいずれであるか。 `local_request = 1` の場合はローカルで、それ以外の場合は GR です。

例 : 1、0

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライスの名前

例 : `session`

db_records_softdelete_total

説明 : `purgeOnEval` が設定されているため、ソフト削除/消去状態になっている DB のレコードの総数

サンプルクエリ : `sum(avg(db_records_softdelete_total{notify=\"1\"}))by(notify)`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライスの名前

例 : `session`

ラベル :

- ラベル : `notify`

ラベルの説明 : `purgeOnNotify` が設定されているかどうか。 1 は `purgeOnNotify=true` を示し、それ以外の場合は 0 です。

例 1

db_records_total

説明：db のレコードの総数。次のメトリックを実現できます。

1. レコードの総数：クエリ：
`sum(avg(db_records_total{namespace=\"$namespace\",session_type=\"total\",appInstanceId=\"0\"})by(systemId,cdl_slice))`
2. スライスワイズレコード数：クエリ：
`sum(avg(db_records_total{namespace=\"$namespace\",session_type=\"total\",appInstanceId=\"0\"})by(systemId,cdl_slice))by(cdl_slice)`
3. システム ID ベースの数：
`sum(avg(db_records_total{namespace=\"$namespace\",session_type=\"total\",appInstanceId=\"0\"})by(systemId,cdl_slice))by(systemId)`
4. セッションタイプによってグループ化されたセッション：クエリ：
`avg(db_records_total{namespace=\"$namespace\",session_type!=\"total\"}) by (session_type)`

サンプルクエリ：

```
sum(avg(db_records_total{namespace=\"$namespace\",session_type=\"total\",appInstanceId=\"0\"})by(systemId,cdl_slice))
```

ラベル：

- ラベル：db

ラベルの説明：DB 名

例：session

ラベル：

- ラベル：session_type

ラベルの説明：データに保存されているセッションタイプ

例：GX、RX、total

ラベル：

- ラベル：systemId

ラベルの説明：システムの ID

例：1、2

ラベル：

- ラベル：cdl_slice

ラベルの説明：論理 CDL スライスの名前

例：session

ラベル：

- ラベル：appInstanceId

ラベルの説明：レコードでアプリケーションによって入力されたアプリケーションインスタンス ID。

例 1

dpapp_internal_requests_total

説明：内部 DP アプリケーション要求の総数

サンプルクエリ：`sum(dpapp_internal_requests_total)by(operation)`

ラベル：

- ラベル：db

ラベルの説明：DB 名

例：session

ラベル：

- ラベル：operation

ラベルの説明：DB 操作のタイプ

例：RemoteBulkRead、RemoteBulkReadIndexing、GetChecksumRemoteSlot

ラベル：

- ラベル：errorCode

ラベルの説明：DB 応答の errorCode

例：0、1401

ラベル：

- ラベル：cdl_slice

ラベルの説明：論理 CDL スライスの名前

例：session

duplicate_slot_records_deleted

説明：重複するスロットデータが見つかったために削除されたスロットレコードの総数

サンプルクエリ：`duplicate_slot_records_deleted`

ラベル：

- ラベル：errorCode

ラベルの説明：DB 応答の errorCode

例：0、502

ラベル：

- ラベル：cdl_slice

ラベルの説明：論理 CDL スライスの名前

例：session

find_no_record_total

説明：レコードが返送されない検索要求の総数

サンプルクエリ：`sum(find_no_record_total)by(not_found_in,operation)`

ラベル：

- ラベル：db

ラベルの説明：DB 名

例：session

ラベル：

- ラベル：operation

ラベルの説明：DB 操作のタイプ

例：FindByUk、FindTagsByUk、Find

ラベル：

- ラベル：not_found_in

ラベルの説明：インデックスまたはスロットでデータが見つからないかどうか

例：index、slot

ラベル：

- ラベル：cdl_slice

ラベルの説明：論理 CDL スライスの名前

例：session

findall_records_bucket

説明：応答で送信されたレコードの数にグループ化できる、受信した findAll 要求の総数

サンプルクエリ：`sum(irate(findall_records_bucket[5m]))by(bucket)`

ラベル：

- ラベル：bucket

ラベルの説明：レコード数によってグループ化されたバケット

例：=0、<=10、<=20、<=50、<=100、>100

ラベル：

- ラベル：cdl_slice

ラベルの説明：論理 CDL スライスの名前

例：session

index_init_sync_duration_seconds

説明：インデックスが起動時にローカルピアおよびリモートピアと同期するのに要した時間

サンプルクエリ：`sum(index_init_sync_duration_seconds)by(shardId,instance_id)`

ラベル：

- ラベル：db

ラベルの説明：DB 名

例：session

ラベル：

- ラベル：shardId

ラベルの説明：シャード ID

例：1、2

ラベル：

- ラベル：systemId

ラベルの説明：システムの ID

例：1、2

index_rebalanced_keys_total

説明：再調整されたインデックスキーの総数

サンプルクエリ：`index_rebalanced_keys_total`

ラベル：

- ラベル：cdl_slice

ラベルの説明：論理 CDL スライスの名前

例：session

ラベル：

- ラベル：shardId

ラベルの説明：シャード ID

例：1、2

indexing_audit_deleted_keys_total

説明：インデックス監査中に削除された一意のキーとプライマリキーの総数

サンプルクエリ：

`sum(irate(indexing_audit_deleted_keys_total{errorCode=\"0\",key_type=\"unique\"}[5m]))by(shardId,instance_id)`

ラベル :

- ラベル : db

ラベルの説明 : DB 名

例 : session

ラベル :

- ラベル : shardId

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : key_type

ラベルの説明 : キーのタイプ

例 : primary、unique

ラベル :

- ラベル : errorCode

ラベルの説明 : DB 応答の errorCode

例 : 302、403

indexing_audit_duration_seconds

説明 : インデックス作成監査の実行に要した合計時間

サンプルクエリ : `indexing_audit_duration_seconds{pod=~\".*\"}`

indexing_audit_total

説明 : インデックス作成監査が実行された合計回数

サンプルクエリ : `indexing_audit_total{pod=~\".*\"}`

indexing_is_leader

説明 : インデックス作成はリーダーまたはフォロワーです

サンプルクエリ : `indexing_is_leader`

ラベル :

- ラベル : db

ラベルの説明 : DB 名

例 : session

ラベル :

- ラベル : shardId

ラベルの説明 : シャード ID

indexing_kafka_replication_delay_seconds

説明 : インデックス内の Kafka からのインデックスのレプリケーションにおける合計遅延

サンプルクエリ :

```
sum(irate(indexing_kafka_replication_delay_seconds[5m]))by(shardId,instance_id)
```

ラベル :

- ラベル : db

ラベルの説明 : DB 名

例 : session

ラベル :

- ラベル : shardId

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : origin_instance_id

ラベルの説明 : kafka 要求の発信元であるインデックスインスタンス ID。

例 : 1.1、1.2

ラベル :

- ラベル : systemId

ラベルの説明 : システムの ID

例 : 1、2

indexing_operation_duration_seconds

説明 : cdl ep からインデックス アプリケーションに送信されたインデックス作成操作の応答に要した時間

サンプルクエリ : `sum(irate(indexing_operation_duration_seconds{errorCode=\"0\"}[5m])) by (operation)`

ラベル :

- ラベル : db

ラベルの説明 : DB 名

例 : session

ラベル :

- ラベル : operation

ラベルの説明 : DB 操作のタイプ

例 : Create、Update、Delete、GetByPk、GetByUk

ラベル :

- ラベル : errorCode

ラベルの説明 : DB 応答の errorCode

例 : 0、404、500、503

ラベル :

- ラベル : cdl_slice

ラベルの説明 : 論理 CDL スライスの名前

例 : session

indexing_operation_total

説明 : cdl ep からインデックス アプリケーションに送信されたインデックス作成操作の総数

サンプルクエリ : `sum(irate(indexing_operation_total{errorCode=\"0\"}[5m])) by (operation)`

ラベル :

- ラベル : db

ラベルの説明 : DB 名

例 : session

ラベル :

- ラベル : operation

ラベルの説明 : DB 操作のタイプ

例 : Create、Update、Delete、GetByPk、GetByUk

ラベル :

- ラベル : errorCode

ラベルの説明 : DB 応答の errorCode

例 : 0、404、500、503

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライスの名前

例 : `session`

indexing_overwrites_total

説明 : インデックスレコードが上書きされているインデックス作成設定操作の総数

サンプルクエリ : `sum(indexing_overwrites_total)by(key_type,shardId)`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : `shardId`

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : `key_type`

ラベルの説明 : キーのタイプ

例 : `primary`、`unique`

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライスの名前

例 : `session`

indexing_records_total

説明 : インデックス作成におけるレコードの総数

サンプルクエリ : `indexing_records_total{pod=~\".*\"}`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : session

ラベル :

- ラベル : shardId

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : cdl_slice

ラベルの説明 : 論理 CDL スライス の 名前

例 : session

indexing_requests_duration_seconds

説明 : 受信したインデックス作成要求の応答に要した時間

サンプルクエリ :

```
sum(irate(indexing_requests_duration_seconds{errorCode=\"0\",isKafka=\"1\"}[5m])) by (operation)
```

ラベル :

- ラベル : db

ラベルの説明 : DB 名

例 : session

ラベル :

- ラベル : operation

ラベルの説明 : DB 操作のタイプ

例 : Set、Delete

ラベル :

- ラベル : shardId

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : errorCode

ラベルの説明 : DB 応答の errorCode

例 : 0、404、1408

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライスの名前

例 : `session`

ラベル :

- ラベル : `isKafka`

ラベルの説明 : 要求が `kafka` と `GRPC` のどちらからのものか。 `isKafka = 1` の場合、その要求は `kafka` からのものです

例 : 1、0

indexing_requests_total

説明 : インデックスポッドで受信した要求の総数

サンプルクエリ : `sum(irate(indexing_requests_total{errorCode=\"0\",isKafka=\"1\"}[5m]))`
by (operation)

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : `operation`

ラベルの説明 : DB 操作のタイプ

例 : `Set`、`Delete`

ラベル :

- ラベル : `shardId`

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : `errorCode`

ラベルの説明 : DB 応答の `errorCode`

例 : 0、404、1408

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライス の名前

例 : `session`

ラベル :

- ラベル : `isKafka`

ラベルの説明 : 要求が `kafka` と `GRPC` のどちらからのものか。 `isKafka = 1` の場合、その要求は `kafka` からのものです

例 : `1`、`0`

`inmemory_indexing_operation_duration_seconds`

説明 : `cdl-ep` から `cdl-index` ポッドへの要求に対する応答に要した合計時間

サンプルクエリ :

```
sum(inmemory_indexing_operation_duration_seconds{errorCode!="0"})by(operation)
```

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : `operation`

ラベルの説明 : DB 操作のタイプ

例 : `Get`、`Multi`

ラベル :

- ラベル : `shardId`

ラベルの説明 : シャード ID

例 : `1`、`2`

ラベル :

- ラベル : `instanceId`

ラベルの説明 : インスタンス ID

例 : `1`、`2`

ラベル :

- ラベル : `errorCode`

ラベルの説明 : DB 応答の `errorCode`

例 : 0、404

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライスの名前

例 : `session`

`inmemory_indexing_operation_total`

説明 : `cdl-ep` から `cdl-index` ポッドへの操作の総数

サンプルクエリ : `sum(inmemory_indexing_operation_total) by (operation, shardId)`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : `operation`

ラベルの説明 : DB 操作のタイプ

例 : `Get`、`Multi`

ラベル :

- ラベル : `shardId`

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : `instanceId`

ラベルの説明 : インスタンス ID

例 : 1、2

ラベル :

- ラベル : `errorCode`

ラベルの説明 : DB 応答の `errorCode`

例 : 0、404

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライス の名前

例 : `session`

kafka_connection_status

説明 : Kafka 接続ステータス

サンプルクエリ : `kafka_connection_status`

ラベル :

- ラベル : `topic`

ラベルの説明 : Kafka トピック名

例 : `kv.kafka.shard.1.1.1`

ラベル :

- ラベル : `shardId`

ラベルの説明 : シャード ID

例 : 1、2

kafka_producer_downtime_op_total

説明 : Kafkaプロデューサが使用できず、ダウンタイムキャッシュに追加された場合の操作の総数

サンプルクエリ : `sum(kafka_producer_downtime_op_total) by (pod,reason)`

ラベル :

- ラベル : `operation`

ラベルの説明 : 失敗した操作

例 : Set、Delete

ラベル :

- ラベル : `reason`

ラベルの説明 : その操作が失敗した理由

例 : `error`、`queue_full`

ラベル :

- ラベル : `success`

ラベルの説明：ダウンタイムキャッシュへの追加が成功したか失敗したか

例：0、1

kafka_producer_pending_publish_total

説明：kafka へのパブリッシュが保留されているメッセージの総数

サンプルクエリ：kafka_producer_pending_publish_total{pod=~\".*\"}

kafka_producer_republished_total

説明：Kafka プロデューサによって再パブリッシュされた要求の総数

サンプルクエリ：kafka_producer_republished_total

ラベル：

- ラベル：operation

ラベルの説明：CDL Kafka の操作

例：Delete、Set

ラベル：

- ラベル：shardId

ラベルの説明：シャード ID

例：1、2

ラベル：

- ラベル：topic

ラベルの説明：Kafka トピック名

例：kv.kafka.shard.1.1.1

kafka_producer_requests_duration_seconds

説明：Kafka プロデューサが要求を処理するのに要した合計時間

サンプルクエリ：sum(irate(kafka_producer_requests_duration_seconds[5m])) by (topic)

ラベル：

- ラベル：topic

ラベルの説明：Kafka トピック名

例：kv.kafka.shard.1.1.1

kafka_producer_requests_total

説明：kafka に送信された要求の総数

サンプルクエリ：kafka_producer_requests_total

ラベル：

- ラベル：topic
- ラベルの説明：Kafka トピック名
- 例：kv.kafka.shard.1.1.1

kafka_records_replayed_total

説明：リーダーのスイッチオーバーまたは kafka の再接続が原因で kafka にパブリッシュされたレコードの合計数

サンプルクエリ：

```
sum(kafka_records_replayed_total{reason=\"leader_switchover\"})by(shardId,instance_id)
```

ラベル：

- ラベル：db
- ラベルの説明：DB 名
- 例：session

ラベル：

- ラベル：shardId
- ラベルの説明：シャード ID
- 例：1、2

ラベル：

- ラベル：reason
- ラベルの説明：kafka レコードを再生する理由
- 例：leader_switchover、kafka_reconnection

notification_ep_connection_total

説明：CDL から通知エンドポイントへの接続の総数

サンプルクエリ：notification_ep_connection_total

ラベル：

- ラベル：cdl_slice
- ラベルの説明：論理 CDL スライスの名前

例 : session

ラベル :

- ラベル : appId

ラベルの説明 : アプリケーション インスタンス ID

例 1

notification_streaming_enabled

説明 : CDL から通知エンドポイントへのストリーミング接続ステータス。ストリーミングが有効になっている場合、値は 1 です。

サンプルクエリ : notification_streaming_enabled

overwritten_index_records_deleted

説明 : インデックスでの一意のキーの上書き/重複が原因で削除されたレコードの総数

サンプルクエリ : overwritten_index_records_deleted

ラベル :

- ラベル : errorCode

ラベルの説明 : 削除のための DB 応答の errorCode

例 : 0、502

ラベル :

- ラベル : cdl_slice

ラベルの説明 : 論理 CDL スライスの名前

例 : session

ラベル :

- ラベル : prefix

ラベルの説明 : 古いレコードを検出した一意のキープレフィックスパターン

例 : uk1

overwritten_index_records_skipped

説明 : キューがいっぱいであるために処理されなかった古いレコードの総数

サンプルクエリ : overwritten_index_records_skipped

ラベル :

- ラベル : action

ラベルの説明：古いレコードに対して実行される予定だったアクション

例：delete、notify

ラベル：

- ラベル：cdl_slice

ラベルの説明：論理 CDL スライスの名前

例：session

ラベル：

- ラベル：prefix

ラベルの説明：古いレコードを検出した一意のキープレフィックスパターン

例：uk1

records_notification_duration_seconds

説明：通知エンドポイントに送信した通知に要した時間

サンプルクエリ：`sum(irate(records_notification_duration_seconds[5m])) by (shardId,instance_id,notification_type)`

ラベル：

- ラベル：db

ラベルの説明：DB 名

例：session

ラベル：

- ラベル：notification_type

ラベルの説明：通知のタイプ

例：TIMER_EXPIRED、RECORD_CONFLICT、BULK_TASK_NOTIFICATION

ラベル：

- ラベル：shardId

ラベルの説明：シャード ID

例：1、2

ラベル：

- ラベル：errorCode

ラベルの説明：DB 応答の errorCode

例：0、1406

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライス の名前

例 : `session`

records_notification_retry_count

説明 : スロットアプリケーションによる通知の再試行の総数

サンプルクエリ : `sum(irate(records_notification_retry_count[5m])) by (shardId,instance_id)`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : `shardId`

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライス の名前

例 : `session`

records_notification_total

説明 : 通知エンドポイントに送信した通知の総数

サンプルクエリ : `sum(irate(records_notification_total[5m])) by (shardId,instance_id,notification_type)`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : `notification_type`

ラベルの説明 : 通知のタイプ

例：TIMER_EXPIRED、RECORD_CONFLICT、BULK_TASK_NOTIFICATION

ラベル：

- ラベル：shardId

ラベルの説明：シャード ID

例：1、2

ラベル：

- ラベル：errorCode

ラベルの説明：DB 応答の errorCode

例：0、1406

ラベル：

- ラベル：cdl_slice

ラベルの説明：論理 CDL スライスの名前

例：session

remote_requests_dropped_total

説明：ドロップされたリモート要求の総数

サンプルクエリ：remote_requests_dropped_total

ラベル：

- ラベル：operation

ラベルの説明：DB 操作のタイプ

例：Create、Update、Delete、UpdateFlags

ラベル：

- ラベル：reason

ラベルの説明：リモート要求をドロップする理由

例：queue_full

remote_site_connection_status

説明：CDL エンドポイントからリモートサイトへの cdl-ep 接続数

サンプルクエリ：sum(remote_site_connection_status)by (pod,systemId)

ラベル：

- ラベル：systemId

ラベルの説明：システムの ID

例：1、2

remote_site_connections_total

説明：エンドポイントポッドごとに設定されたリモートサイト接続の総数

サンプルクエリ：remote_site_connections_total

ラベル：

- ラベル：systemId

ラベルの説明：リモートサイトの systemId ID

例：1、2

slot_checksum_mismatch_total

説明：チェックサムの不一致の総数

サンプルクエリ：sum(irate(slot_checksum_mismatch_total[5m]))by(slot_shard_id)

ラベル：

- ラベル：db

ラベルの説明：DB 名

例：session

ラベル：

- ラベル：slot_shard_id

ラベルの説明：スロットシャード ID

例：1、2

slot_geo_replication_requests_duration_seconds

説明：スロット Geo レプリケーションの応答の送信に要した時間

サンプルクエリ：

sum(irate(slot_geo_replication_requests_duration_seconds[5m]))by(systemId,operation)

ラベル：

- ラベル：systemId

ラベルの説明：システムの ID

例：1、2

ラベル：

- ラベル : operation

ラベルの説明 : DB 操作のタイプ

例 : CREATE、DELETE、UPDATE、UPDATEFLAGS

ラベル :

- ラベル : errorCode

ラベルの説明 : DB 応答の errorCode

例 : 0、503

slot_geo_replication_requests_total

説明 : スロット Geo レプリケーションの要求の総数

サンプルクエリ : `sum(irate(slot_geo_replication_requests_total[5m]))by(systemId,operation)`

ラベル :

- ラベル : systemId

ラベルの説明 : システムの ID

例 : 1、2

ラベル :

- ラベル : operation

ラベルの説明 : DB 操作のタイプ

例 : CREATE、DELETE、UPDATE、UPDATEFLAGS

ラベル :

- ラベル : errorCode

ラベルの説明 : DB 応答の errorCode

例 : 0、503

slot_init_sync_duration_seconds

説明 : 起動時にスロットがローカルピアおよびリモートピアと同期するために要した時間

サンプルクエリ : `sum(slot_init_sync_duration_seconds)by(shardId,instance_id)`

ラベル :

- ラベル : db

ラベルの説明 : DB 名

例 : session

ラベル :

- ラベル : shardId

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : systemId

ラベルの説明 : システムの ID

例 : 1、2

slot_operation_duration_seconds

説明 : cdl ep からスロットアプリケーションに送信された操作の応答に要した時間

サンプルクエリ :

```
sum(irate(slot_operation_duration_seconds{errorCode=\"0\",local_request=\"1\"}[5m])) by (operation)
```

ラベル :

- ラベル : db

ラベルの説明 : DB 名

例 : session

ラベル :

- ラベル : operation

ラベルの説明 : DB 操作のタイプ

例 : Create、Update、Delete、Find、UpdateFlags

ラベル :

- ラベル : slot_shard_id

ラベルの説明 : スロットシャード ID。メトリックの詳細度が実稼働の場合、空の文字列が保持されます

例 : 1、2

ラベル :

- ラベル : slot_instance_id

ラベルの説明 : スロットインスタンス ID。メトリックの詳細度が実稼働の場合、空の文字列が保持されます

例 : 1、2

ラベル :

- ラベル : `errorCode`

ラベルの説明 : DB 応答の `errorCode`

例 : 0、104、105

ラベル :

- ラベル : `local_request`

ラベルの説明 : DB 要求がローカルまたは GR のいずれであるか。 `local_request = 1` の場合はローカルで、それ以外の場合は GR です。

例 : 1、0

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライスの名前

例 : `session`

slot_operation_total

説明 : `cdl ep` からスロットアプリケーションに送信された操作の総数

サンプルクエリ : `sum(irate(slot_operation_total(errorCode=\"0\",local_request=\"1\")[5m]))`
`by (operation)`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : `operation`

ラベルの説明 : DB 操作のタイプ

例 : Create、Update、Delete、Find、UpdateFlags

ラベル :

- ラベル : `slot_shard_id`

ラベルの説明 : スロットシャード ID。メトリックの詳細度が実稼働の場合は、空の文字列です

例 : 1、2

ラベル :

- ラベル : `slot_instance_id`

ラベルの説明 : スロットインスタンス ID。メトリックの詳細度が実稼働の場合は、空の文字列です

例 : 1、2

ラベル :

- ラベル : `errorCode`

ラベルの説明 : DB 応答の `errorCode`

例 : 0、104、105

ラベル :

- ラベル : `local_request`

ラベルの説明 : DB 要求がローカルまたは GR のいずれであるか。 `local_request = 1` の場合はローカルで、それ以外の場合は GR です。

例 : 1、0

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライスの名前

例 : `session`

slot_purged_sessions_duration_seconds

説明 : 次の評価タイマーが切れ、`purge=true` であることによる、スロットでのセッションの消去に要した時間

サンプルクエリ :

```
sum(irate(slot_purged_sessions_duration_seconds{errorCode=\"0\"}[5m]))by(shardId,instance_id)
```

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : `shardId`

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : `errorCode`

ラベルの説明 : DB 応答の `errorCode`

例 : 0、501、508

例 : インデックスレコードの取得に失敗しました。無効なスライス名を受信しました

ラベル :

- ラベル : `notify`

ラベルの説明 : `purgeOnNotify` が設定されているかどうか。1 は `purgeOnNotify=true` を示し、それ以外の場合は 0 です。

例 1

slot_purged_sessions_total

説明 : 次の評価タイマーが切れ、`purge=true` であることにより、スロットで消去されたセッションの総数

サンプルクエリ :

```
sum(irate(slot_purged_sessions_total{errorCode=\"0\"}[5m]))by(shardId,instance_id)
```

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : `shardId`

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : `errorCode`

ラベルの説明 : DB 応答の `errorCode`

例 : 0、501、508

ラベル :

- ラベル : `notify`

ラベルの説明 : `purgeOnNotify` が設定されているかどうか。1 は `purgeOnNotify=true` を示し、それ以外の場合は 0 です。

例 1

slot_reconciled_records_total

説明：調整されたレコードの総数

サンプルクエリ：

```
sum(slot_reconciled_records_total)by(systemId,slot_shard_id,slot_instance_id)
```

ラベル：

- ラベル：db

ラベルの説明：DB 名

例：session

ラベル：

- ラベル：slot_shard_id

ラベルの説明：スロットシャード ID

例：1、2

ラベル：

- ラベル：slot_instance_id

ラベルの説明：スロットインスタンス ID

例：1、2

ラベル：

- ラベル：systemId

ラベルの説明：システムの ID

例：1、2

ラベル：

- ラベル：operation

ラベルの説明：DB 操作のタイプ

例：Create、Delete

slot_reconciliation_duration_seconds

説明：調整の実行に要した合計時間

サンプルクエリ：sum(slot_reconciliation_duration_seconds{isError=\"0\"})by(slot_shard_id)

ラベル：

- ラベル : db

ラベルの説明 : DB 名

例 : session

ラベル :

- ラベル : slot_shard_id

ラベルの説明 : スロットシャード ID

例 : 1、2

ラベル :

- ラベル : isError

ラベルの説明 : 調整中にエラーが発生したかどうか。isError = 1 の場合、エラーが発生しています

例 : 0、1

slot_reconciliation_total

説明 : チェックサムの不一致によってトリガーされた調整の総数

サンプルクエリ : `sum(slot_reconciliation_total{isError=\"0\"})by(slot_shard_id)`

ラベル :

- ラベル : db

ラベルの説明 : DB 名

例 : session

ラベル :

- ラベル : slot_shard_id

ラベルの説明 : スロットシャード ID

例 : 1、2

ラベル :

- ラベル : isError

ラベルの説明 : 調整中にエラーが発生したかどうか。isError = 1 の場合、エラーが発生しています

例 : 0、1

slot_records_size_total

説明：スロット内のレコードの合計サイズ（バイト単位）

サンプルクエリ：sum(slot_records_size_total)

ラベル：

- ラベル：db

ラベルの説明：DB 名

例：session

ラベル：

- ラベル：shardId

ラベルの説明：シャード ID

例：1、2

ラベル：

- ラベル：cdl_slice

ラベルの説明：論理 CDL スライスの名前

例：session

slot_records_total

説明：スロット内のレコードの総数

サンプルクエリ：sum(slot_records_total{session_type\"total\"}) by (pod)

ラベル：

- ラベル：db

ラベルの説明：DB 名

例：session

ラベル：

- ラベル：shardId

ラベルの説明：シャード ID

例：1、2

ラベル：

- ラベル：cdl_slice

ラベルの説明：論理 CDL スライスの名前

例：session

ラベル :

- ラベル : `session_type`

ラベルの説明 : データに保存されているセッションタイプ

例 : GX、RX、total

ラベル :

- ラベル : `systemId`

ラベルの説明 : システムの ID

例 : 1、2

ラベル :

- ラベル : `bucket`

ラベルの説明 : サイズ別にグループ化されたバケット

例 : $\leq$ 1kb、2kb、4kb、8kb

ラベル :

- ラベル : `appInstanceId`

ラベルの説明 : レコードでアプリケーションによって入力されたアプリケーションインスタンス ID。

例 1

slot_requests_duration_second

説明 : スロットアプリケーションで受信した要求の応答に要した時間

サンプルクエリ : `sum(irate(slot_requests_duration_seconds{errorCode\"0\"}[5m])) by (errorCode)`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : session

ラベル :

- ラベル : `operation`

ラベルの説明 : DB 操作のタイプ

例 : Get、Create、Delete

ラベル :

- ラベル : `shardId`

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : `errorCode`

ラベルの説明 : DB 応答の `errorCode`

例 : 0、1406

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライスの名前

例 : `session`

slot_requests_total

説明 : スロットアプリケーションで受信した要求の総数

サンプルクエリ : `sum(irate(slot_requests_total{errorCode=\"0\"}[5m])) by (operation)`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : `operation`

ラベルの説明 : DB 操作のタイプ

例 : Get、Create、Delete

ラベル :

- ラベル : `shardId`

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : `errorCode`

ラベルの説明 : DB 応答の `errorCode`

例 : 0、1406

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライスの名前

例 : `session`

slot_stale_record_duration_seconds

説明 : スロットが古いスロットレコードを処理するのに要した時間

サンプルクエリ : `slot_stale_record_duration_seconds`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : `delete`

ラベルの説明 : 古いレコードが削除するために送信されているか、またはスキップされたかを確認します。`delete=1` の場合は削除するために送信されており、それ以外の場合はスキップされています。

例 : 1、0

ラベル :

- ラベル : `shardId`

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : `errorCode`

ラベルの説明 : DB 応答の `errorCode`

例 : 0、502

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライスの名前

例 : `session`

ラベル :

- ラベル : `reason`

ラベルの説明 : 古いレコードの削除の理由

例 : `find_all_notify`、`stale_check_enabled`

slot_stale_record_total

説明 : 処理された古いスロットレコードの削除の総数

サンプルクエリ : `slot_stale_record_total`

ラベル :

- ラベル : `db`

ラベルの説明 : DB 名

例 : `session`

ラベル :

- ラベル : `delete`

ラベルの説明 : 古いレコードが削除するために送信されているか、またはスキップされたかを確認します。`delete=1` の場合は削除するために送信されており、それ以外の場合はスキップされています。

例 : 1、0

ラベル :

- ラベル : `shardId`

ラベルの説明 : シャード ID

例 : 1、2

ラベル :

- ラベル : `errorCode`

ラベルの説明 : DB 応答の `errorCode`

例 : 0、502

ラベル :

- ラベル : `cdl_slice`

ラベルの説明 : 論理 CDL スライスの名前

例 : `session`

ラベル :

- ラベル : `reason`

ラベルの説明 : 古いレコードの削除の理由

例 : `find_all_notify`、`stale_check_enabled`



第 Ⅰ 部

CDL 設定モードのコマンドリファレンス

- [CDL 設定モードのコマンドリファレンス \(127 ページ\)](#)



第 4 章

CDL 設定モードのコマンドリファレンス

- [cdl \(129 ページ\)](#)
- [cdl actions backup \(130 ページ\)](#)
- [cdl actions restore \(131 ページ\)](#)
- [cdl api \(131 ページ\)](#)
- [cdl datastore \(132 ページ\)](#)
- [cdl datastore actions remote-index-sync start \(133 ページ\)](#)
- [cdl datastore actions remote-index-sync start map-id \(134 ページ\)](#)
- [cdl datastore actions remote-index-sync start slice-name \(134 ページ\)](#)
- [cdl datastore actions remote-index-sync status \(134 ページ\)](#)
- [cdl datastore clear sessions \(135 ページ\)](#)
- [cdl datastore clear sessions filter \(135 ページ\)](#)
- [cdl datastore clear sessions in-filter \(136 ページ\)](#)
- [cdl datastore endpoint \(137 ページ\)](#)
- [cdl datastore endpoint settings \(139 ページ\)](#)
- [cdl datastore features index-overwrite-detection unique-keys-prefix \(139 ページ\)](#)
- [cdl datastore features instance-aware-notification \(140 ページ\)](#)
- [cdl datastore features instance-aware-notification system-id \(140 ページ\)](#)
- [cdl datastore features overload-protection \(141 ページ\)](#)
- [cdl datastore features slot-stale-session-detection \(143 ページ\)](#)
- [cdl datastore index \(144 ページ\)](#)
- [cdl datastore kafka \(145 ページ\)](#)
- [cdl datastore kafka describe \(147 ページ\)](#)
- [cdl datastore kafka describe consumer-groups \(148 ページ\)](#)
- [cdl datastore kafka describe topics \(148 ページ\)](#)
- [cdl datastore kafka external-ip \(148 ページ\)](#)
- [cdl datastore kafka label-config \(149 ページ\)](#)
- [cdl datastore kafka list \(149 ページ\)](#)
- [cdl datastore kafka list consumer-groups \(150 ページ\)](#)
- [cdl datastore kafka list topics \(150 ページ\)](#)

- [cdl datastore kafka log-compaction](#) (150 ページ)
- [cdl datastore kafka reset-offsets](#) (151 ページ)
- [cdl datastore kafka reset-offsets consumer-group](#) (151 ページ)
- [cdl datastore kafka ssl-settings](#) (152 ページ)
- [cdl datastore metrics](#) (152 ページ)
- [cdl datastore mirror-maker](#) (153 ページ)
- [cdl datastore overload-protection](#) (153 ページ)
- [cdl datastore reset-offsets-dry-run](#) (154 ページ)
- [cdl datastore reset-offsets-dry-run consumer-group](#) (154 ページ)
- [cdl datastore session features remote-site-connection-monitoring](#) (155 ページ)
- [cdl datastore show indexes](#) (155 ページ)
- [cdl datastore show sessions count detailed](#) (156 ページ)
- [cdl datastore show sessions count detailed filter](#) (157 ページ)
- [cdl datastore show sessions count detailed in-filter](#) (158 ページ)
- [cdl datastore show sessions count summary](#) (158 ページ)
- [cdl datastore show sessions count summary filter](#) (159 ページ)
- [cdl datastore show sessions count summary in-filter](#) (160 ページ)
- [cdl datastore show sessions detailed](#) (161 ページ)
- [cdl datastore show sessions detailed filter](#) (162 ページ)
- [cdl datastore show sessions detailed in-filter](#) (163 ページ)
- [cdl datastore show sessions summary](#) (163 ページ)
- [cdl datastore show sessions summary filter](#) (165 ページ)
- [cdl datastore show sessions summary in-filter](#) (165 ページ)
- [cdl datastore show status](#) (166 ページ)
- [cdl datastore slot](#) (167 ページ)
- [cdl datastore slot notification](#) (168 ページ)
- [cdl datastore slot notification metrics](#) (170 ページ)
- [cdl datastore zookeeper](#) (171 ページ)
- [cdl label-config](#) (172 ページ)
- [cdl label-config endpoint](#) (173 ページ)
- [cdl label-config index map](#) (173 ページ)
- [cdl label-config slot map](#) (174 ページ)
- [cdl logging](#) (174 ページ)
- [cdl logging logger](#) (175 ページ)
- [cdl remote-site](#) (176 ページ)
- [cdl remote-site db-endpoint](#) (176 ページ)
- [cdl remote-site kafka-server](#) (177 ページ)
- [cdl ssl-config](#) (177 ページ)
- [cdl ssl-config certs](#) (178 ページ)

cdl

次世代の Cisco Data Layer (CDL) を設定します。

コマンド モード

[Exec] > [Global Configuration]

構文の説明

```
cdl { datastore datastore_name { cluster-id cluster_id | { { endpoint {
external-ip ipv4_address | external-port port_number } | index { map
index_partitions | write-factor write_factor } | slot { map slot_partitions |
notification { host host_name | limit notification_tps | port
notification_server_port } | write-factor write_factor } } { go-debug go_debug
| patch-url patch_url | replica ha_instances } | geo-remote-site list } } |
deployment-model { large | small } | enable-geo-replication { false |
true } | kafka { enable-JMX-metrics { false | true } | enable-persistence
{ false | true } | external-ip { ipv4_address port_number } | replica
ha_instances | retention-size data_retention_size | retention-time
data_retention_time | storage data_storage_size } | logging { {
default-log-level | logger cdl_module_name level default_log_level } | node-type
k8s_node_label_value | remote-site system_id { db-endpoint { host
db_endpoint_hostname | port db_endpoint_port_number } | kafka-server { host port
} } | system-id system_id | zookeeper { data-storage-size data_storage_size
| { enable-JMX-metrics | enable-persistence } { false | true } |
log-storage-size log_storage_size | replica ha_instances } } }
```

deployment-model *deployment_model*

k8s デプロイメントのリソース要件を指定します。デフォルト値は `large` です。注：実稼働環境では、常に大規模導入モデルを使用する必要があります。

次のいずれかである必要があります。

- `large`
- `small`

enable-geo-replication { false | true }

Geo レプリケーションを有効にするか無効にするかを指定します。

次のいずれかである必要があります。

- `false`
- `true`

デフォルト値は `false` です。

node-type *k8s_node_label_value*

アフィニティの K8s ノードのラベル値を指定します。

0 ～ 64 文字の文字列である必要があります。

デフォルト値：

system-id *system_id*

デプロイメントシステム ID を指定します。

整数である必要があります。

デフォルト値は 1 です。

使用上のガイドライン

このコマンドを使用して、デプロイメントの CDL パラメータを設定します。CDL コンテナが設定されていない場合、CDL チャートは展開されません。

cdl actions backup

バックアッププロセスをトリガーします。

コマンド モード

[Exec] > [Global Configuration]

構文の説明

```
cdl actions backup start [ concurrency-factor concurrency_factor |
created-before timestamp | slice-name slice_name ]
```

concurrency-factor *concurrency_factor*

スロットから同時にデータを取得してバックアップを取ります。デフォルトでは、一度にクエリされるスロットは 1 つだけです。

concurrency_factor は、1 ～ 8 の範囲の整数である必要があります。

created-before *timestamp*

このタイムスタンプより前に作成されたレコードのバックアップを取ります。タイムスタンプは CCYY-MM-DDTHH:MM:SS 形式にする必要があります。

デフォルトでは、使用可能なすべてのデータがバックアップのためにクエリされます。

slice-name *slice_name*

指定したスライス名のバックアップを取ります。ここでは、設定したスライス名のみ渡すことができます。

デフォルトでは、すべてのスライス名のバックアップを取ります。

使用上のガイドライン

このコマンドを使用するとバックアッププロセスがトリガーされますが、プロセスが完了するまで待機はしません。作成されたダンプファイル名とともに成功または失敗のメッセージを返します。バックアッププロセスが完了すると、復元プロセスのために出力のダンプファイルを収集または参照できます。このファイルは、**cdl-api-0** ポッドを実行しているノードの `/data/<namespace>/data-cdl-api-0/<file-name.dump>` に配置できます。

このコマンドを実行すると、これにより **CDL** バックアッププロセスがトリガーされます。よろしいですか? という警告が表示されます。**yes** と入力して、CDL バックアッププロセスのトリガーを続行します。

cdl actions restore

復元プロセスをトリガーします。

コマンドモード

[Exec] > [Global Configuration]

構文の説明

```
cdl actions restore { start file filename | status }
```

start file *filename*

このキーワードの設定は必須です。

ファイル名には **.dump** 拡張子が付いている必要があります。ファイルは、**cdl-api-0** ポッドを実行しているノードの、ホストパス **/data/<namespace>/data-cdl-api-0/<file-name.dump>** に存在する必要があります。

status

復元プロセスのステータスを次のいずれかのオプションで表示します。

次のいずれかである必要があります。

- **idle | completed** : CDL 復元プロセスが完了した、または進行中ではない
- **in-progress** : CDL 復元プロセスが進行中である

使用上のガイドライン

このコマンドを使用すると復元プロセスがトリガーされますが、プロセスが完了するまで待機はしません。成功または失敗のメッセージが返されます。このコマンドでファイルパラメータを設定する必要があります

このコマンドを実行すると、これにより **CDL** 復元プロセスがトリガーされ、既存のデータが上書きされます。よろしいですか? という警告が表示されます。**yes** と入力して、CDL 復元プロセスのトリガーを続行します。

cdl api

CDL API 設定を有効にします。

コマンドモード

[Exec] > [Global Configuration]

構文の説明

```
cdl api [ deploy { false | true } | cpu-limit cpu_limit | go-max-procs  
cpu_cores | memory-limit memory_limit | storage storage_size ]
```

deploy { false | true }

API ポッドのデプロイメントを有効または無効にします。

デフォルト : **false**

cpu-limit *cpu_limit*

最大 CPU 制限を指定します。*cpu_limit* は 1000 ～ 8000 の範囲の整数である必要があります。

デフォルト : 4000

go-max-procs *cpu_cores*

CPU コアの最大数を指定します。*cpu_cores* は 1 ～ 32 の範囲の整数である必要があります。

デフォルト : 4

memory-limit *memory_limit*

許可される最大メモリを Gi 単位で指定します。*memory_limit* は 1 ～ 8 の範囲の整数である必要があります。

デフォルト : 4

storage *storage_size*

バックアップストレージを GB 単位で指定します。*storage_size* は 1 ～ 200 の範囲の整数である必要があります。

デフォルト : 90

使用上のガイドライン このコマンドを使用して、CDL API 設定にアクセスします。CDL API を設定するために、CDL データストアが設定されていることを確認します。

cdl datastore

展開するデータストアのリストを設定します。

コマンドモード

[Exec] > [Global Configuration]

構文の説明

```
cdl datastore datastore_name { advanced-config-profile profile_name |
cluster-id cluster_id | copies-per-node copies_per_node | label-config
label_config | { { endpoint { external-ip ipv4_address | external-port
port_number } | index { map index_partitions | write-factor write_factor } |
slot { map slot_partitions | notification { host host_name | limit
notification_tps | port notification_server_port } | write-factor write_factor }
} { go-debug go_debug | patch-url patch_url | replica ha_instances } } |
geo-remote-site list }
```

advanced-config-profile *profile_name*

詳細設定パラメータのプロファイル名を指定します。

cluster-id *cluster_id*

K8s クラスタ ID を指定します。

1 ～ 10 の範囲の整数である必要があります。

デフォルト値は 1 です。

find-by-nuk-prefixes *key_prefix*

findbyNonUniqueKey API をサポートするように事前設定された非一意のキープレフィックスのリストを指定します

1 ～ 20 文字の文字列である必要があります。

geo-remote-site *list*

Geo レプリケーションサイトのリストを指定します。

label-config *label_config*

ラベル設定名を指定します。

slice-names *slice_names_list*

スライス名のリストを指定します。

1 ～ 16 文字の文字列である必要があります。

デフォルト値は session です。

datastore_name

データストアの名前を指定します。

1 ～ 16 文字の文字列である必要があります。

使用上のガイドライン このコマンドを使用して、展開するデータストアのリストを設定します。
このコマンドで最大 10 の要素を設定できます。

cdl datastore actions remote-index-sync start

インデックスのリモートピアとの同期を開始します。

コマンドモード

EXEC

構文の説明

actions remote-index-sync start

使用上のガイドライン インデックスをリモートピアとともに調整します。このコマンドを使用して、インデックスのリモートピアとの同期を開始します。

cdl datastore actions remote-index-sync start map-id

インデックスマップ ID のリストのリモート同期をトリガーします。

コマンドモード EXEC

構文の説明 `actions remote-index-sync map-id map index_name`

`map index_map`

インデックスマップのリモートインデックス同期のトリガーを指定します。

整数である必要があります。

使用上のガイドライン このコマンドを使用して、インデックスマップ ID のリストのリモート同期をトリガーします。デフォルトでは、すべてのインデックスマップがトリガーされます。

このコマンドで最大 5 つの要素を設定できます。

cdl datastore actions remote-index-sync start slice-name

スライス名のリストのリモート同期をトリガーします。

コマンドモード EXEC

構文の説明 `actions remote-index-sync slice-name slice slice_name`

`slice slice_name`

指定すると、スライス名のリモートインデックス同期をトリガーします。

文字列である必要があります。

使用上のガイドライン このコマンドを使用して、スライス名のリストのリモート同期をトリガーします。デフォルトでは、すべてのスライス名がトリガーされます。

cdl datastore actions remote-index-sync status

リモートピアとまだ同期しているインデックスインスタンスを表示します。

コマンドモード EXEC

構文の説明 `cdl show status`

使用上のガイドライン このコマンドを使用して、リモートピアとまだ同期しているインデックスインスタンスのカンマ区切りリストを表示します。

cdl datastore clear sessions

すべてのデータストアセッションデータをクリアします。

コマンドモード EXEC

構文の説明 `cdl clear sessions [ db-name db_name | filter { condition query_expression key query_string } | map-id map_id ]`

db-name db_name

クエリする DB の名前を指定します。

1 ～ 16 文字の文字列である必要があります。

デフォルト値は session です。

map-id map_id

マップからデータをクエリするためのマップ ID を指定します。

0 ～ 1024 の範囲の整数である必要があります。

デフォルト値は 0 です。

slice-name slice_name

クエリするスライス名を指定します。

1 ～ 16 文字の文字列である必要があります。

デフォルト値は session です。

使用上のガイドライン このコマンドを使用して、すべてのデータストアセッションデータをクリアします。これにより、一部またはすべてのレコードが完全に削除される可能性があることに注意してください。

cdl datastore clear sessions filter

クエリに使用する AND フィルタパラメータを設定します。

コマンドモード EXEC

構文の説明 `filter { condition query_expression key query_string } | map-id map_id | limit number_of_records | purge-on-eval { false | true } | next-eval-start-time next_eval_start_time | next-eval-end-time next_eval_end_time`

condition *query_expression*

AND フィルタでクエリ式を指定します。

次のいずれかである必要があります。

- **contains**
- **ends-with**
- **match**
- **not-contains**
- **not-ends-with**
- **not-match**
- **not-starts-with**
- **starts-with**

key *query_string*

クエリ文字列を指定します。

0 ～ 512 文字の文字列である必要があります。

使用上のガイドライン このコマンドを使用して、クエリに使用するフィルタパラメータを設定します。

cdl datastore clear sessions in-filter

クエリに使用する OR フィルタパラメータを設定します。

コマンドモード

EXEC

構文の説明

```
filter { condition query_expression key query_string } | map-id map_id | limit number_of_records | purge-on-eval { false | true } | next-eval-start-time next_eval_start_time | next-eval-end-time next_eval_end_time
```

condition *query_expression*

OR フィルタでクエリ式を指定します。

次のいずれかである必要があります。

- **contains**
- **ends-with**
- **match**
- **not-contains**
- **not-ends-with**

- **not-match**
- **not-starts-with**
- **starts-with**

key *query_string*

クエリ文字列を指定します。

0 ～ 512 文字の文字列である必要があります。

使用上のガイドライン このコマンドを使用して、クエリに使用するフィルタパラメータを設定します。

cdl datastore endpoint

エンドポイントポッドのパラメータを設定します。

コマンド モード

[Exec] > [Global Configuration]

構文の説明

```
cdl datastore datastore_name endpoint { go-debug go_debug_variables | patch-url
patch_url | replica ha_instances }
```

copies-per-node *copies_per_node*

ノードごとに cdl-endpoint ポッドのコピーを指定します。

1 ～ 4 の範囲の整数である必要があります。

デフォルト値は 1 です。

cpu-limit *max_cpu*

許容される最大 CPU をミリ CPU 単位で指定します。

200 ～ 32000 の範囲の整数である必要があります。

デフォルト値は 4000 です。

cpu-request *min_cpu*

必要な最小 CPU をミリ CPU 単位で指定します。

100 ～ 16000 の範囲の整数である必要があります。

デフォルト値は 2000 です。

external-ip *ipv4_address*

サービスを外部に公開する IPv4 アドレスのリストを指定します。

external-ip *ipv6_address*

サービスを外部に公開する IPv6 アドレスのリストを指定します。

external-port *port_number*

外部トラフィックに公開するポート番号を指定します。

デフォルト値は 8882 です。

go-debug *go_debug_variables*

GO デバッグ変数のカンマ区切りリストを指定します。

文字列である必要があります。

go-max-procs *max_cpu*

同時に使用する CPU コアの最大数を指定します。

1 ～ 32 の範囲の整数である必要があります。

デフォルト値は 4 です。

memory-limit *max_memory*

許容される最大メモリをメガバイト (MB) 単位で指定します。

512 ～ 131072 の範囲の整数である必要があります。

デフォルト値は 131072 です。

memory-request *min_memory*

必要な最小メモリをメガバイト (MB) 単位で指定します。

256 ～ 65536 の範囲の整数である必要があります。

デフォルト値は 256 です。

patch-url *patch_url*

データストア アプリケーションのパッチをダウンロードするための URL を指定します。

文字列である必要があります。

replica *ha_instances*

作成する HA インスタンスの数を指定します。

1 ～ 16 の範囲の整数である必要があります。

デフォルト値は 2 です。

使用上のガイドライン このコマンドを使用して、エンドポイントポッドのパラメータを設定します。

cdl datastore endpoint settings

このコマンドは廃止されました。CD エンドポイントのパラメータを設定します。

コマンド モード

[Exec] > [Global Configuration]

構文の説明

```
settings index-timeout-ms index_timeout slot-timeout-ms slot_timeout
remote-ep-connection-count remote_ep_connection_count
```

index-timeout-ms *index_timeout*

このキーワードは廃止されます。インデックス操作のタイムアウト期間をミリ秒単位で指定します。

25 ～ 5000 の範囲の整数である必要があります。

デフォルト値は 200 です。

remote-ep-connection-count *remote_ep_connection_count*

このキーワードは廃止されます。GRPC レプリケーションのリモートサイト CDL エンドポイントへの接続数を指定します。

1 ～ 20 の範囲の整数である必要があります。

デフォルト値は 4 です。

slot-timeout-ms *slot_timeout*

このキーワードは廃止されます。スロット操作のタイムアウト時間をミリ秒単位で指定します。

25 ～ 5000 の範囲の整数である必要があります。

デフォルト値は 375 です。

使用上のガイドライン

このコマンドは廃止されました。このコマンドを使用して、CDL エンドポイントパラメータを設定します。

cdl datastore features index-overwrite-detection unique-keys-prefix

CDL オーバーロード防止設定を行います。

コマンド モード

[Exec] > [Global Configuration]

構文の説明

```
index-overwrite-detection unique-keys-prefix { action index_overwrite_action
| key-prefix key_prefix | max-tps max_tps | queue-size queue_size }
```

action *index_overwrite_action*

インデックスの上書き検出時に実行するアクションを指定します。

次のいずれかである必要があります。

- **delete-record**
- **log-record**
- **notify-record**

key-prefix *key_prefix*

上書き検出の一意のキーのプレフィックスパターンを指定します。

文字列である必要があります。

使用上のガイドライン

インデックスキーは、検出設定を上書きします。このコマンドを使用して、インデックス上書き検出用の **uniqueKey** プレフィックスのリストと、検出時に実行するアクションを設定します。

cdl datastore features instance-aware-notification

Geo レプリケーションのインスタンス認識設定を行います。

コマンドモード

[Exec] > [Global Configuration]

構文の説明

```
features instance-aware-notification enable { false | true }
```

enable { false | true }

スライスのインスタンス状態チェックを有効にするかどうかを指定します。

次のいずれかである必要があります。

- **false**
- **true**

デフォルト値は **false** です。

使用上のガイドライン

このコマンドを使用して、Geo レプリケーションのインスタンス認識設定を行います。

cdl datastore features instance-aware-notification system-id

sliceNames への systemID のマッピングを設定します。

コマンドモード

[Exec] > [Global Configuration]

構文の説明

system-id is *system_id* **slice-names** *slice_names*

id *system_id*

システム/サイト ID を指定します。

整数である必要があります。

slice-names *slice_names*

systemID に関連付けられているスライス名のリストを指定します。

使用上のガイドライン

このコマンドを使用して、sliceNames への systemID のマッピングを設定します。

cdl datastore features overload-protection

CDL オーバーロード防止設定を行います。

コマンドモード

[Exec] > [Global Configuration]

構文の説明

```
cdl datastore overload-protection disable { false | true } |
critical-alert-threshold critical_alert_threshold | hard-limit-percentage
hard_limit_percentage | index-record-capacity-count index_record_capacity_count
| major-alert-threshold major_alert_threshold | slot-record-capacity-bytes
slot_record_capacity | slot-record-capacity-count slot_record_capacity_count
```

enable { false | true }

オーバーロード防止を無効にするか有効にするかを指定します。

次のいずれかである必要があります。

- false
- true

デフォルト値は false です。

index-max-record-count *index_max_record_count*

インデックスシャードに保存できるレコードの最大数を指定します。

整数である必要があります。

100 ~ 1000 または 100000 ~ 100000000 の範囲の整数である必要があります

デフォルト値は 60000000 です。

enable { false | true }

オーバーロード防止を無効にするか有効にするかを指定します。

次のいずれかである必要があります。

- false
- true

デフォルト値は false です。

index-max-record-count *index_max_record_count*

インデックスシャードに保存できるレコードの最大数を指定します。

整数である必要があります。

slot-max-record-count *slot_max_record_count*

スロットシャードに保存できるレコードの最大数を指定します。

整数である必要があります。

100 ～ 1000 または 100000 ～ 100000000 の範囲の整数である必要があります

デフォルト値は 2500000 です。

critical-alert-threshold *critical_alert_threshold*

CDL がアラート `cdlOverloaded-critical` をトリガーする必要があるしきい値をパーセンテージで指定します。major-alert-threshold は critical-alert-threshold よりも小さくする必要があります。

40 ～ 100 の範囲の整数である必要があります。

デフォルト値は 90 です。

enable { false | true }

オーバーロード防止を無効にするか有効にするかを指定します。

次のいずれかである必要があります。

- false
- true

デフォルト値は false です。

hard-limit-percentage *hard_limit_percentage*

ソフト制限に加えて、追加のキャパシティのパーセンテージを指定します。

0 ～ 10 の範囲の整数である必要があります。

デフォルト値は 5 です。

index-max-record-count *index_max_record_count*

インデックスシャードに保存できるレコードの最大数を指定します。

整数である必要があります。

major-alert-threshold *major_alert_threshold*

CDL がアラート `cdlOverloaded-major` をトリガーする必要があるしきい値をパーセンテージで指定します。`major-alert-threshold` は `critical-alert-threshold` よりも小さくする必要があります。

40 ～ 100 の範囲の整数である必要があります。

デフォルト値は 80 です。

slot-max-record-count *slot_max_record_count*

スロットシャードに保存できるレコードの最大数を指定します。

整数である必要があります。

slot-max-size *slot_max_size*

スロットシャードの最大サイズをメガバイト単位で指定します。

1024 ～ 98304 の範囲の整数である必要があります。

デフォルト値は 16384 です。

使用上のガイドライン このコマンドを使用して、CDL オーバーロード防止設定を行います。

cdl datastore features slot-stale-session-detection

古いスロットレコードの検出パラメータを設定します。

コマンド モード

[Exec] > [Global Configuration]

構文の説明

slot-stale-session-detection disable-auto-deletion { false | true }

disable-auto-deletion { false | true }

古いスロットレコードの自動削除を無効にするか有効にするかを指定します。

次のいずれかである必要があります。

- false
- true

デフォルト値は false です。

notification-retry-count *notification_retry_count*

アプリケーションから更新を受信せずにアプリケーションに送信するタイマー切れ通知の最小再試行回数を指定します。「notification-retry-count」の回数を超えても更新を受信しない場合、CDL はスロットレコードが古いかどうかのチェックに進みます。

3 ～ 10 の範囲の整数である必要があります。

デフォルト値は 3 です。

使用上のガイドライン このコマンドを使用して、古いスロットレコードの検出パラメータを設定します。

cdl datastore index

インデックスポッドのパラメータを設定します。

コマンドモード

[Exec] > [Global Configuration]

構文の説明

```
cdl datastore datastore_name index { map index_partitions | write-factor
write_factor | go-debug go_debug | patch-url patch_url | replica ha_instances }
```

cpu-limit *max_cpu*

許容される最大 CPU をミリ CPU 単位で指定します。

200 ～ 32000 の範囲の整数である必要があります。

デフォルト値は 4000 です。

cpu-request *min_cpu*

必要な最小 CPU をミリ CPU 単位で指定します。

100 ～ 16000 の範囲の整数である必要があります。

デフォルト値は 2000 です。

go-debug *go_debug_variables*

GO デバッグ変数のカンマ区切りリストを指定します。

文字列である必要があります。

go-max-procs *max_cpu*

同時に使用する CPU コアの最大数を指定します。

1 ～ 32 の範囲の整数である必要があります。

デフォルト値は 4 です。

map *index_partitions*

インデックス用に作成するパーティションの数を指定します。

1 ～ 1024 の範囲の整数である必要があります。

デフォルト値は 1 です。

memory-limit *max_memory*

許容される最大メモリをメガバイト (MB) 単位で指定します。

512 ～ 131072 の範囲の整数である必要があります。

デフォルト値は 131072 です。

memory-request *min_memory*

必要な最小メモリをメガバイト (MB) 単位で指定します。

256 ～ 65536 の範囲の整数である必要があります。

デフォルト値は 256 です。

patch-url *patch_url*

データストア アプリケーションのパッチをダウンロードするための URL を指定します。

文字列である必要があります。

replica *replicas*

作成する HA インスタンスの数を指定します。

1 ～ 2 の範囲の整数である必要があります。

デフォルト値は 2 です。

write-factor *write_factor*

このキーワードは廃止されました。正常な応答の前にインデックスに書き込まれるコピーの数を指定します。

0 ～ 2 の範囲の整数である必要があります。

デフォルト値は 2 です。

使用上のガイドライン このコマンドを使用して、インデックスポッドのパラメータを設定します。

cdl datastore kafka

Geo レプリケーション用の Kafka を設定します。

コマンドモード

[Exec] > [Global Configuration]

構文の説明

```
cdl kafka { enable-JMX-metrics { false | true } | enable-persistence {
  false | true } | external-ip { ipv4_address port_number } | replica
ha_instances | retention-size retention_size | retention data_retention_period |
  storage data_storage_size | log-level { debug | error | fatal | info |
trace | warn } }
```

enable-JMX-metrics { false | true }

JMX メトリックを有効にするか無効にするかを指定します。

次のいずれかである必要があります。

- false
- true

デフォルト値は true です。

enable-persistence { false | true }

Kafka データの永続性を有効にするか無効にするかを指定します。

次のいずれかである必要があります。

- false
- true

デフォルト値は false です。

log-level

Kafka のロギングレベルを指定します。ロギングレベルには、[debug]、[error]、[fatal]、[info]、[trace]、[warn] があります。ここで指定された変更は、Zookeeper ポッドのローリング再起動をトリガーします。

次のいずれかである必要があります。

- debug
- error
- fatal
- info
- trace
- warn

partition *number_of_partitions*

トピックごとのパーティションの数を指定します。

1 ～ 1024 の範囲の整数である必要があります。

デフォルト値は 1 です。

replica *ha_instances*

作成する HA インスタンスの数を指定します。

1 ～ 16 の範囲の整数である必要があります。

デフォルト値は 2 です。

デフォルト値は 2 です。

retention-size *retention_size*

このキーワードは廃止されます。データ保持サイズを MB 単位で指定します。

1024 ～ 1048576 の範囲の整数である必要があります。

デフォルト値は 5120 です。

retention-time *data_retention_period*

データ保持期間を時間単位で指定します。

1 から 168 の範囲の整数である必要があります。

デフォルト値は 12 です。

storage *data_storage_size*

Kafka データのストレージサイズを GB 単位で指定します。

1 ～ 64 の範囲の整数である必要があります。

デフォルト値は 20 です。

to-earliest

指定すると、並列ポッド管理ポリシーが有効になります。

次のいずれかである必要があります。

- false
- true

デフォルト値は false です。

total-retention-size *total_retention_size*

データの合計保持サイズを MB 単位で指定します。

10240 ～ 1048576 の範囲の整数である必要があります。

デフォルト値は 65536 です。

使用上のガイドライン このコマンドを使用して、Geo レプリケーション用の Kafka を設定します。

cdl datastore kafka describe

Kafka コンシューマグループについて説明します。デフォルトでは、すべての Kafka コンシューマグループについて説明されています。

コマンドモード EXEC

構文の説明

```
cdl kafka describe { consumer-groups name group_name | topics {
include-internal-topics | name kafka-topic_name } }
```

使用上のガイドライン

このコマンドを使用して、Kafka コンシューマグループについて説明します。

cdl datastore kafka describe consumer-groups

Kafka コンシューマグループについて説明します。

コマンドモード

EXEC

構文の説明

```
cdl kafka describe consumer-groups name group_name
```

group name *group_name*

Kafka コンシューマグループ名を指定します。

パターン [a-zA-Z0-9._-]+ の文字列である必要があります。

使用上のガイドライン

このコマンドを使用して、すべての Kafka コンシューマグループを一覧表示します。デフォルトの動作では、すべてのコンシューマグループが記述されます。

cdl datastore kafka describe topics

Kafka のトピックについて説明します。デフォルトでは、内部以外のすべてのトピックについて説明します。

コマンドモード

EXEC

構文の説明

```
cdl kafka describe topics { include-internal-topics | name kafka_topic_name
}
```

include-internal-topics

内部の Kafka トピックを含めるかどうかを指定します。

topic *kafka-topic_name*

Kafka トピック名を指定します。

パターン [a-zA-Z0-9._-]+ の文字列である必要があります。

使用上のガイドライン

このコマンドを使用して、Kafka トピックを説明します。

cdl datastore kafka external-ip

外部サーバーに公開する IP を設定します。

コマンドモード	[Exec] > [Global Configuration]
構文の説明	<pre>cdl kafka external-ip <i>ipv4_address</i> port <i>port_number</i> ssl-port <i>ssl_port_number</i></pre> ip <i>IPv4_address</i> Kafkaサーバーがクライアントのトラフィックをリッスンするための IPv4 アドレスを指定します。 port <i>port_number</i> 外部トラフィックに公開するポートアドレスを指定します。 ssl-port <i>ssl-port_address</i> 外部トラフィックに公開する ssl-port アドレスを指定します。
使用上のガイドライン	このコマンドを使用して、外部サーバーに公開する IP を設定します。

cdl datastore kafka label-config

Kafka ポッドのラベル設定を行います。

コマンドモード	[Exec] > [Global Configuration]
構文の説明	<pre>cdl kafka label-config key <i>label_key</i> value <i>label_value</i></pre> key <i>label_key</i> ラベルキーを指定します。 文字列である必要があります。 value <i>label_value</i> ラベル値を指定します。 文字列である必要があります。

使用上のガイドライン このコマンドを使用して、Kafka ポッドのラベル設定を行います。

cdl datastore kafka list

すべての kafka トピックを表示します。デフォルトでは、リストに内部トピックは含まれていません。

コマンドモード	EXEC
---------	------

構文の説明

```
cdl kafka list { consumer-groups name group_name | topics {
include-internal-topics | name kafka-topic_name } }
```

使用上のガイドライン

このコマンドを使用して、すべての Kafka トピックを表示します。

cdl datastore kafka list consumer-groups

すべての Kafka コンシューマグループを表示します。

コマンドモード

EXEC

構文の説明

```
cdl kafka list consumer-groups name group_name
```

使用上のガイドライン

このコマンドを使用して、すべての Kafka コンシューマグループを表示します。

cdl datastore kafka list topics

すべての Kafka トピックを表示します。

コマンドモード

EXEC

構文の説明

```
cdl kafka list topics include-internal-topics
```

```
include-internal-topics
```

内部 Kafka トピックを含めるかどうかを指定します。

使用上のガイドライン

このコマンドを使用して、すべての Kafka トピックを表示します。

cdl datastore kafka log-compaction

Kafka ログの圧縮パラメータを設定します。

コマンドモード

[Exec] > [Global Configuration]

構文の説明

```
cdl kafka log-compaction disable { false | true } delete-retention-ms
delete_retention_period
```

```
delete-retention-ms delete_retention_period
```

ログの圧縮されたトピックの削除を保持する期間をミリ秒単位で指定します。

60000 ～ 172800000 の範囲の整数である必要があります。

デフォルト値は 43200000 です。

disable { false | true }

Kafka ログの圧縮を無効にするか有効にするかを指定します。

次のいずれかである必要があります。

- false
- true

デフォルト値は false です。

使用上のガイドライン このコマンドを使用して、Kafka ログの圧縮パラメータを設定します。

cdl datastore kafka reset-offsets

Kafka コンシューマグループのすべてのトピックのオフセットをリセットします。コンシューマグループのオフセットは、グループが非アクティブの場合にのみ設定できます。

コマンドモード

EXEC

構文の説明

```
cdl kafka reset-offsets { consumer-group name group_name | to-earliest name group_name | to-latest name group_name }
```

使用上のガイドライン

このコマンドを使用して、Kafka コンシューマグループのすべてのトピックのオフセットをリセットします。

cdl datastore kafka reset-offsets consumer-group

すべてのトピックの Kafka コンシューマグループのオフセットをリセットします。

コマンドモード

EXEC

構文の説明

```
cdl kafka reset-offsets consumer-group name group_name
```

consumer-group name group_name

Kafka コンシューマグループを指定します。

パターン [a-zA-Z0-9._-]+ の文字列である必要があります。

to-earliest

指定すると、オフセットが最も古いものにリセットされます。

to-latest

指定すると、オフセットが最新のものにリセットされます。

使用上のガイドライン このコマンドを使用して、すべてのトピックの Kafka コンシューマグループのオフセットをリセットします。

cdl datastore kafka ssl-settings

Kafka SSL の設定を行います。

コマンドモード [Exec] > [Global Configuration]

構文の説明

```
cdl kafka ssl-settings enable-ssl { false | true }
disable-host-name-verification { false | true }
```

disable-host-name-verification { false | true }

Kafka ブローカ間の SSL 通信時にホスト名の検証を無効にするか有効にするかを指定します。
次のいずれかである必要があります。

- **false**
- **true**

デフォルト値は **false** です。

enable-ssl { false | true }

Kafka ブローカ間の SSL 通信を無効にするか有効にするかを指定します。
次のいずれかである必要があります。

- **false**
- **true**

デフォルト値は **false** です。

使用上のガイドライン このコマンドを使用して、Kafka SSL の設定を行います。

cdl datastore metrics

メトリックの詳細レベルを設定します。

コマンドモード [Exec] > [Global Configuration]

構文の説明

```
cdl datastore datastore_name { metrics { verbosity { debug | production } }
```

使用上のガイドライン このコマンドを使用して、メトリックの詳細レベルを設定します。

cdl datastore mirror-maker

Kafka MirrorMaker を設定します。

コマンドモード

[Exec] > [Global Configuration]

構文の説明

```
cdl mirror-maker { enable-JMX-metrics { false | true } | log-level {
debug | error | fatal | info | trace | warn } }
```

enable-JMX-metrics { false | true }

JMX メトリックを有効にするか無効にするかを指定します。

次のいずれかである必要があります。

- **false**
- **true**

デフォルト値は **true** です。

log-level { debug | error | fatal | info | trace | warn }

Kafka MirrorMaster のロギングレベルを指定します。注：この変更により、MirrorMaker ポッドの再起動がトリガーされます。

次のいずれかである必要があります。

- **debug**
- **error**
- **fatal**
- **info**
- **trace**
- **warn**

使用上のガイドライン このコマンドを使用して、Kafka MirrorMaker を設定します。

cdl datastore overload-protection

このコマンドは廃止されました。CDL オーバーロード防止設定を行います。

コマンドモード

[Exec] > [Global Configuration]

構文の説明

```
cdl datastore overload-protection disable { false | true } |
critical-alert-threshold critical_alert_threshold | hard-limit-percentage
hard_limit_percentage | index-record-capacity-count index_record_capacity_count
```

```
| major-alert-threshold major_alert_threshold | slot-record-capacity-bytes
slot_record_capacity | slot-record-capacity-count slot_record_capacity_count
```

disable { false | true }

このキーワードは廃止されます。オーバーロード防止を無効にするか有効にするかを指定します。

次のいずれかである必要があります。

- **false**
- **true**

デフォルト値は **false** です。

使用上のガイドライン このコマンドは廃止されました。このコマンドを使用して、CDL オーバーロード防止設定を行います。

cdl datastore reset-offsets-dry-run

コンシューマグループで変更を実行せずに **reset-offsets** の結果を表示します。

コマンドモード EXEC

構文の説明 **cdl kafka reset-offsets-dry-run { consumer-group name *group_name* | to-earliest name *group_name* | to-latest name *group_name* }**

使用上のガイドライン このコマンドを使用して、コンシューマグループで変更を実行せずに **reset-offsets** の結果を表示します。

cdl datastore reset-offsets-dry-run consumer-group

すべてのトピックの Kafka コンシューマグループのオフセットをリセットします。

コマンドモード EXEC

構文の説明 **cdl kafka reset-offsets-dry-run consumer-group name *group_name***

consumer-group name *group_name*

Kafka コンシューマグループを指定します。

パターン **[a-zA-Z0-9._-]+** の文字列である必要があります。

to-earliest

指定すると、オフセットが最も古いものにリセットされます。

to-earliest

指定すると、オフセットが最新のものにリセットされます。

使用上のガイドライン このコマンドを使用して、すべてのトピックの Kafka コンシューマグループのオフセットをリセットします。

cdl datastore session features remote-site-connection-monitoring

CDL エンドポイントのリモートサイト接続（レプリケーション接続と内部運用接続）を設定します。

コマンド モード [Exec] > [Global Configuration]

構文の説明 `features remote-site-connection-monitoring enable { false | true }`

`enable { false | true }`

リモートサイトのモニタリングを有効にするか無効にするかを指定します。この機能は、デフォルトで有効化されています。

次のいずれかである必要があります。

- `false`
- `true`

デフォルト値 : `true`

使用上のガイドライン このコマンドを使用して、CDL エンドポイントのリモートサイト接続を設定します。

cdl datastore show indexes

セッションインデックス情報を表示します。

コマンド モード EXEC

構文の説明 `show indexes all-replicas { false | true } | db-name db_name | key key | limit max_indexes | map-id map_id ]`

`all-replicas { false | true }`

すべてのレプリカからインデックスを取得するかどうかを指定します。

次のいずれかである必要があります。

- `false`

- true

デフォルト値は false です。

db-name *db_name*

クエリする DB の名前を指定します。

1 ～ 16 文字の文字列である必要があります。

デフォルト値は session です。

key *key*

キー：クエリ値（uk/pk）を指定します。

0 ～ 512 文字の文字列である必要があります。

limit *max_indexes*

表示するインデックスの最大数を指定します。

1 ～ 1000 の範囲の整数である必要があります。

map-id *map_id*

インデックスが保存されているマップからデータをクエリするためのマップIDを指定します。

0 ～ 1024 の範囲の整数である必要があります。

slice-name *slice_name*

クエリするスライス名を指定します。

1 ～ 16 文字の文字列である必要があります。

デフォルト値は session です。

使用上のガイドライン このコマンドを使用して、セッションインデックス情報を表示します。

cdl datastore show sessions count detailed

セッション数の詳細情報を表示します。

コマンドモード

EXEC

構文の説明

```
cdl show sessions count detailed [ db-name db_name | filter { condition
query_expression key query_string } | limit number_of_records | map-id map_id ]
```

db-name *db_name*

クエリする DB の名前を指定します。

1 ～ 16 文字の文字列である必要があります。

デフォルト値は `session` です。

map-id *map_id*

マップからデータをクエリするためのマップ ID を指定します。

0 ～ 1024 の範囲の整数である必要があります。

デフォルト値は 0 です。

slice-name *slice_name*

クエリするスライス名を指定します。

1 ～ 16 文字の文字列である必要があります。

デフォルト値は `session` です。

使用上のガイドライン このコマンドを使用して、セッション数の詳細情報を表示します。

cdl datastore show sessions count detailed filter

クエリに使用する AND フィルタパラメータを設定します。

コマンド モード EXEC

構文の説明

```
filter { condition query_expression key query_string } | map-id map_id | limit
number_of_records | purge-on-eval { false | true } | next-eval-start-time
next_eval_start_time | next-eval-end-time next_eval_end_time
```

condition *query_expression*

AND フィルタでクエリ式を指定します。

次のいずれかである必要があります。

- **contains**
- **ends-with**
- **match**
- **not-contains**
- **not-ends-with**
- **not-match**
- **not-starts-with**
- **starts-with**

key *query_string*

クエリ文字列を指定します。

0 ～ 512 文字の文字列である必要があります。

使用上のガイドライン このコマンドを使用して、クエリに使用するフィルタパラメータを設定します。

cdl datastore show sessions count detailed in-filter

クエリに使用する OR フィルタパラメータを設定します。

コマンドモード EXEC

構文の説明

```
filter { condition query_expression key query_string } | map-id map_id | limit
number_of_records | purge-on-eval { false | true } | next-eval-start-time
next_eval_start_time | next-eval-end-time next_eval_end_time
```

condition *query_expression*

OR フィルタでクエリ式を指定します。

次のいずれかである必要があります。

- **contains**
- **ends-with**
- **match**
- **not-contains**
- **not-ends-with**
- **not-match**
- **not-starts-with**
- **starts-with**

key *query_string*

クエリ文字列を指定します。

0 ～ 512 文字の文字列である必要があります。

使用上のガイドライン このコマンドを使用して、クエリに使用するフィルタパラメータを設定します。

cdl datastore show sessions count summary

サマリーセッション数の情報を表示します。

コマンドモード

EXEC

構文の説明

```
cdl show sessions count summary [ db-name db_name | filter { condition query_expression key query_string } | limit number_of_records | map-id map_id ]
```

db-name *db_name*

クエリする DB の名前を指定します。

1 ～ 16 文字の文字列である必要があります。

デフォルト値は `session` です。

map-id *map_id*

マップからデータをクエリするためのマップ ID を指定します。

0 ～ 1024 の範囲の整数である必要があります。

デフォルト値は 0 です。

slice-name *slice_name*

クエリするスライス名を指定します。

1 ～ 16 文字の文字列である必要があります。

デフォルト値は `session` です。

使用上のガイドライン

このコマンドを使用して、サマリーセッション数の情報を表示します。

cdl datastore show sessions count summary filter

クエリに使用する AND フィルタパラメータを設定します。

コマンドモード

EXEC

構文の説明

```
filter { condition query_expression key query_string } | map-id map_id | limit number_of_records | purge-on-eval { false | true } | next-eval-start-time next_eval_start_time | next-eval-end-time next_eval_end_time
```

condition *query_expression*

AND フィルタでクエリ式を指定します。

次のいずれかである必要があります。

- **contains**
- **ends-with**
- **match**
- **not-contains**

- **not-ends-with**
- **not-match**
- **not-starts-with**
- **starts-with**

key *query_string*

クエリ文字列を指定します。

0 ～ 512 文字の文字列である必要があります。

使用上のガイドライン このコマンドを使用して、クエリに使用するフィルタパラメータを設定します。

cdl datastore show sessions count summary in-filter

クエリに使用する OR フィルタパラメータを設定します。

コマンド モード

EXEC

構文の説明

```
filter { condition query_expression key query_string } | map-id map_id | limit
number_of_records | purge-on-eval { false | true } | next-eval-start-time
next_eval_start_time | next-eval-end-time next_eval_end_time
```

condition *query_expression*

OR フィルタでクエリ式を指定します。

次のいずれかである必要があります。

- **contains**
- **ends-with**
- **match**
- **not-contains**
- **not-ends-with**
- **not-match**
- **not-starts-with**
- **starts-with**

key *query_string*

クエリ文字列を指定します。

0 ～ 512 文字の文字列である必要があります。

使用上のガイドライン このコマンドを使用して、クエリに使用するフィルタパラメータを設定します。

cdl datastore show sessions detailed

セッションの詳細をデータとともに表示します。

構文の説明

```
cdl show detailed [ db-name db_name | filter { condition query_expression key query_string } | limit number_of_records | map-id map_id ]
```

コマンドモード

EXEC

db-name *db_name*

クエリする DB の名前を指定します。

1 ～ 16 文字の文字列である必要があります。

デフォルト値は session です。

limit *number_of_records*

表示するレコードの最大数を指定します。

1 ～ 500 の範囲の整数である必要があります。

デフォルト値は 100 です。

map-id *map_id*

マップからデータをクエリするためのマップ ID を指定します。

0 ～ 1024 の範囲の整数である必要があります。

デフォルト値は 0 です。

max-data-size-kb *max_data_size*

レコードごとに表示する最大データサイズを指定します。データサイズがこのパラメータより大きい場合、出力は切り捨てられます。

1 ～ 32 の範囲の整数である必要があります。

デフォルト値は 4 です。

next-eval-end-time *next_eval_end_time*

next-eval-time がこの時間より短いセッション。

next-eval-start-time *next_eval_start_time*

next-eval-time がこの時間より長いセッション。

purge-on-eval { false | true }

purge-on-eval フラグが false または true に設定されているセッション。

次のいずれかである必要があります。

- false
- true

slice-name *slice_name*

クエリするスライス名を指定します。

1 ～ 16 文字の文字列である必要があります。

デフォルト値は session です。

使用上のガイドライン このコマンドを使用して、セッションの詳細をデータとともに表示します。

cdl datastore show sessions detailed filter

クエリに使用する AND フィルタパラメータを設定します。

コマンドモード

EXEC

構文の説明

```
filter { condition query_expression key query_string } | map-id map_id | limit  
number_of_records | purge-on-eval { false | true } | next-eval-start-time  
next_eval_start_time | next-eval-end-time next_eval_end_time
```

condition *query_expression*

AND フィルタでクエリ式を指定します。

次のいずれかである必要があります。

- contains
- ends-with
- match
- not-contains
- not-ends-with
- not-match
- not-starts-with
- starts-with

key *query_string*

クエリ文字列を指定します。

0 ～ 512 文字の文字列である必要があります。

使用上のガイドライン このコマンドを使用して、クエリに使用するフィルタパラメータを設定します。

cdl datastore show sessions detailed in-filter

クエリに使用する OR フィルタパラメータを設定します。

コマンド モード EXEC

構文の説明

```
filter { condition query_expression key query_string } | map-id map_id | limit
number_of_records | purge-on-eval { false | true } | next-eval-start-time
next_eval_start_time | next-eval-end-time next_eval_end_time
```

condition *query_expression*

OR フィルタでクエリ式を指定します。

次のいずれかである必要があります。

- **contains**
- **ends-with**
- **match**
- **not-contains**
- **not-ends-with**
- **not-match**
- **not-starts-with**
- **starts-with**

key *query_string*

クエリ文字列を指定します。

0 ～ 512 文字の文字列である必要があります。

使用上のガイドライン このコマンドを使用して、クエリに使用するフィルタパラメータを設定します。

cdl datastore show sessions summary

データを含まないセッションの詳細を表示します。

構文の説明

```
cdl show sessions summary [ db-name db_name | filter { condition
query_expression key query_string } | limit number_of_records | map-id map_id ]
```

コマンドモード

EXEC

db-name *db_name*

クエリする DB の名前を指定します。

1 ～ 16 文字の文字列である必要があります。

デフォルト値は session です。

limit *number_of_records*

表示するレコードの最大数を指定します。

1 ～ 500 の範囲の整数である必要があります。

デフォルト値は 100 です。

map-id *map_id*

マップからデータをクエリするためのマップ ID を指定します。

0 ～ 1024 の範囲の整数である必要があります。

デフォルト値は 0 です。

max-data-size-kb *max_data_size*

レコードごとに表示する最大データサイズを指定します。データサイズがこのパラメータより大きい場合、出力は切り捨てられます。

1 ～ 32 の範囲の整数である必要があります。

デフォルト値は 4 です。

next-eval-end-time *next_eval_end_time*

next-eval-time がこの時間より短いセッション。

next-eval-start-time *next_eval_start_time*

next-eval-time がこの時間より長いセッション。

purge-on-eval { false | true }

purge-on-eval フラグが false または true に設定されているセッション。

次のいずれかである必要があります。

- false
- true

slice-name *slice_name*

クエリするスライス名を指定します。

1 ～ 16 文字の文字列である必要があります。

デフォルト値は session です。

使用上のガイドライン このコマンドを使用して、データを含まないセッションの詳細を表示します。

cdl datastore show sessions summary filter

クエリに使用する AND フィルタパラメータを設定します。

コマンドモード

EXEC

構文の説明

```
filter { condition query_expression key query_string } | map-id map_id | limit
number_of_records | purge-on-eval { false | true } | next-eval-start-time
next_eval_start_time | next-eval-end-time next_eval_end_time
```

condition *query_expression*

AND フィルタでクエリ式を指定します。

次のいずれかである必要があります。

- **contains**
- **ends-with**
- **match**
- **not-contains**
- **not-ends-with**
- **not-match**
- **not-starts-with**
- **starts-with**

key *query_string*

クエリ文字列を指定します。

0 ～ 512 文字の文字列である必要があります。

使用上のガイドライン このコマンドを使用して、クエリに使用するフィルタパラメータを設定します。

cdl datastore show sessions summary in-filter

クエリに使用する OR フィルタパラメータを設定します。

コマンドモード

EXEC

構文の説明

```
filter { condition query_expression key query_string } | map-id map_id | limit number_of_records | purge-on-eval { false | true } | next-eval-start-time next_eval_start_time | next-eval-end-time next_eval_end_time
```

condition *query_expression*

OR フィルタでクエリ式を指定します。

次のいずれかである必要があります。

- **contains**
- **ends-with**
- **match**
- **not-contains**
- **not-ends-with**
- **not-match**
- **not-starts-with**
- **starts-with**

key *query_string*

クエリ文字列を指定します。

0 ～ 512 文字の文字列である必要があります。

使用上のガイドライン このコマンドを使用して、クエリに使用するフィルタパラメータを設定します。

cdl datastore show status

CDL ステータス情報を表示します。

コマンドモード

EXEC

構文の説明

```
cdl show status [ db-name db_name ]
```

db-name *db_name*

データベースの名前を指定します。

文字列である必要があります。

デフォルト値は `session` です。

使用上のガイドライン このコマンドを使用して、CDL ステータス情報を表示します。

cdl datastore slot

スロットポッドのパラメータを設定します。

コマンドモード

[Exec] > [Global Configuration]

構文の説明

```
cdl datastore datastore_name slot { map slot_partitions | notification { host host_name | limit notification_tps | port notification_server_port } | write-factor write_factor }
```

cpu-limit *max_cpu*

許容される最大 CPU をミリ CPU 単位で指定します。

200 ～ 32000 の範囲の整数である必要があります。

デフォルト値は 4000 です。

cpu-request *min_cpu*

必要な最小 CPU をミリ CPU 単位で指定します。

100 ～ 16000 の範囲の整数である必要があります。

デフォルト値は 2000 です。

go-debug *go_debug_variables*

GO デバッグ変数のカンマ区切りリストを指定します。

文字列である必要があります。

go-max-procs *max_cpu*

同時に使用する CPU コアの最大数を指定します。

1 ～ 32 の範囲の整数である必要があります。

デフォルト値は 4 です。

map *slot_partitions*

スロットに作成するパーティションの数を指定します。

1 ～ 1024 の範囲の整数である必要があります。

デフォルト値は 1 です。

memory-limit *max_memory*

許容される最大メモリをメガバイト（MB）単位で指定します。

512 ～ 131072 の範囲の整数である必要があります。

デフォルト値は 131072 です。

memory-request *min_memory*

必要な最小メモリをメガバイト（MB）単位で指定します。

256 ～ 65536 の範囲の整数である必要があります。

デフォルト値は 256 です。

patch-url *patch_url*

データストア アプリケーションのパッチをダウンロードするための URL を指定します。

文字列である必要があります。

replica *ha_instances*

作成する HA インスタンスの数を指定します。

1 ～ 4 の範囲の整数である必要があります。

デフォルト値は 2 です。

write-factor *write_factor*

正常な応答の前にスロットに書き込まれるコピー数を指定します。

1 ～ 4 の範囲の整数である必要があります。

デフォルト値は 2 です。

使用上のガイドライン このコマンドを使用して、スロットポッドのパラメータを設定します。

cdl datastore slot notification

アプリケーションへの通知設定を行います。

コマンド モード

[Exec] > [Global Configuration]

構文の説明

```
cdl datastore slot notification { connection-count connection_count |
dynamic-provisioning { false | true } | host host_name | limit
notification_tps | max-concurrent-bulk-notifications max_bulk_task_notifications
| notification-retry-seconds notification_retry_interval | port
notification_server_port | remote-system-id remote_system_id }
```

connection-count *connection_count*

アプリケーション通知エンドポイントへの通知接続を指定します。

1 ～ 20 の範囲の整数である必要があります。

デフォルト値は 4 です。

dynamic-provisioning { false | true }

通知エンドポイントをアプリケーションによってプロビジョニングするか、静的設定を使用するかを指定します。true の場合、ダイナミック プロビジョニングが有効になります。

次のいずれかである必要があります。

- false
- true

デフォルト値は false です。

host *host_name*

通知サーバーのホスト名を指定します。

文字列である必要があります。

デフォルト値は datastore-notification-ep です。

include-conflict-data { false | true }

指定すると、元のデータと要求からのデータを DB 競合通知とともに受信する設定が有効になります。

次のいずれかである必要があります。

- false
- true

デフォルト値は false です。

limit *notification_tps*

通知サーバーの 1 秒あたりのトランザクション (TPS) を指定します。

1 ～ 2000 の範囲の整数である必要があります。

デフォルト値は 200 です。

max-concurrent-bulk-notifications *max_notifications*

CDL が同時に処理できる一括タスク通知の最大数を指定します。

1 ～ 20 の範囲の整数である必要があります。

デフォルト値は 1 です。

notification-retry-seconds *notification_retry_interval*

アプリケーションがタイマー切れ通知後に次の評価時間を更新しない場合に、CDL がアプリケーションへの通知を再試行するまでの秒数を指定します。

10 ～ 100 の範囲の整数である必要があります。

デフォルト値は 10 です。

port notification_server_port

通知サーバーのポート番号を指定します。

デフォルト値は 8890 です。

remote-system-id remote_system_id

リモートのシステム ID のリストを指定します。レコードの所有者システム ID がリモート ID のいずれかと一致する場合、CDL から通知が送信されます。

use-stream { false | true }

指定すると、CDL からアプリケーションに通知を送信するためのストリーミングを有効にします。

次のいずれかである必要があります。

- false
- true

デフォルト値は false です。

使用上のガイドライン このコマンドを使用して、アプリケーションへの通知設定を行います。

cdl datastore slot notification metrics

CDL スロットメトリックのパラメータを設定します。

コマンドモード [Exec] > [Global Configuration]

構文の説明 **metrics report-idle-session-type { false | true }**

report-idle-session-type { false | true }

アイドルセッションタイプを報告するかどうかを指定します。

次のいずれかである必要があります。

- false
- true

デフォルト値は false です。

使用上のガイドライン このコマンドを使用して、CDL スロットメトリックのパラメータを設定します。

cdl datastore zookeeper

Geo レプリケーション用の Zookeeper を設定します。

コマンドモード

[Exec] > [Global Configuration]

構文の説明

```
cdl zookeeper { data-storage-size data_storage_size | { enable-JMX-metrics  
| enable-persistence } { false | true } | log-storage-size log_storage_size  
| replica ha_instances | log-level { debug | error | fatal | info | trace  
| warn } }
```

data-storage-size *data_storage_size*

Zookeeper データのストレージサイズを GB 単位で指定します。

1 ～ 64 の範囲の整数である必要があります。

デフォルト値は 20 です。

enable-JMX-metrics { false | true }

JMX メトリックを有効にするか無効にするかを指定します。

次のいずれかである必要があります。

- false
- true

デフォルト値は true です。

enable-persistence { false | true }

Zookeeper データの永続性を有効にするか無効にするかを指定します。

次のいずれかである必要があります。

- false
- true

デフォルト値は false です。

log-level

Zookeeper のロギングレベルを指定します。ロギングレベルには、[debug]、[error]、[fatal]、[info]、[trace]、[warn] があります。ここで指定された変更は、Zookeeper ポッドのローリング再起動をトリガーします。

次のいずれかである必要があります。

- debug
- error

- fatal
- info
- trace
- warn

log-storage-size *log_storage_size*

Zookeeper データログのストレージサイズを GB 単位で指定します。

1 ～ 64 の範囲の整数である必要があります。

デフォルト値は 20 です。

replica *ha_instances*

作成する HA インスタンスの数を指定します。

次のいずれかの整数である必要があります：1、3。

デフォルト値は 3 です。

to-earliest

指定すると、並列ポッド管理ポリシーが有効になります。

次のいずれかである必要があります。

- false
- true

デフォルト値は false です。

使用上のガイドライン このコマンドを使用して、Geo レプリケーション用の Zookeeper を設定します。

cdl label-config

CDL ポッドのラベル設定のリストを設定します。

コマンド モード [Exec] > [Global Configuration]

構文の説明 **cdl label-config** *label_config_name*

label_config_name

ラベル設定名を指定します。

文字列である必要があります。

使用上のガイドライン このコマンドを使用して、CDL ポッドのラベル設定のリストを設定します。

cdl label-config endpoint

エンドポイントのラベル設定を行います。

コマンド モード

[Exec] > [Global Configuration]

構文の説明

endpoint **key** *label_key* **value** *label_value*

key *label_key*

ラベルキーを指定します。

文字列である必要があります。

value *label_value*

ラベル値を指定します。

文字列である必要があります。

使用上のガイドライン

このコマンドを使用して、エンドポイントのラベル設定を行います。

cdl label-config index map

CDL インデックスのマップレベルのラベル設定を行います。

コマンド モード

[Exec] > [Global Configuration]

構文の説明

index map **map-id** *map_id*

key *label_key*

ラベルキーを指定します。

文字列である必要があります。

map-id *map_id*

マップベースのラベル設定を指定します。

整数である必要があります。

value *label_value*

ラベル値を指定します。

文字列である必要があります。

使用上のガイドライン

このコマンドを使用して、CDL インデックスのマップレベルのラベル設定を行います。

cdl label-config slot map

CDL スロットのマップレベルのラベル設定を行います。

コマンドモード [Exec] > [Global Configuration]

構文の説明 **slot map map-id** *map_id*

key *label_key*

ラベルキーを指定します。

文字列である必要があります。

map-id *map_id*

マップベースのラベル設定を指定します。

整数である必要があります。

value *label_value*

ラベル値を指定します。

文字列である必要があります。

使用上のガイドライン このコマンドを使用して、CDL スロットのマップレベルのラベル設定を行います。

cdl logging

CDL ロギングパラメータを設定します。

コマンドモード [Exec] > [Global Configuration]

構文の説明 **cdl logging default-log-level** *default_log_level* **error-stack** { **false** | **true** } **sync-logging** { **false** | **true** }

default-log-level *default_log_level*

CDL モジュールのデフォルトのロギングレベルを指定します。

次のいずれかである必要があります。

- debug
- error
- info
- off
- trace

- **warn**

デフォルト値は **warn** です。

error-stack { false | true }

エラーログと連動するスタックトレースを有効にするか無効にするかを指定します。有効にするには、**true** に設定します。

次のいずれかである必要があります。

- **false**
- **true**

デフォルト値は **false** です。

json-logging { false | true }

json 形式でのロギングを有効にするか無効にするかを指定します。

次のいずれかである必要があります。

- **false**
- **true**

デフォルト値は **false** です。

sync-logging { false | true }

同期ロギングを有効にするか無効にするかを指定します。有効にするには、**true** に設定します。

次のいずれかである必要があります。

- **false**
- **true**

デフォルト値は **false** です。

使用上のガイドライン このコマンドを使用して、CDL ロギングパラメータを設定します。

cdl logging logger

各モジュールのログを設定します。

コマンドモード [Exec] > [Global Configuration]

構文の説明 **cdl logging logger** *cdl_module_name* **level** *default_log_level*

level default_log_level

各モジュールのロギングレベルを指定します。

次のいずれかである必要があります。

- debug
- error
- info
- off
- trace
- warn

cdl_module_name

CDL モジュールの名前を指定します。

文字列である必要があります。

使用上のガイドライン このコマンドを使用して、各モジュールのログを設定します。

cdl remote-site

リモートサイトのリストを設定します。

コマンドモード [Exec] > [Global Configuration]

構文の説明 **cdl remote-site** *system_id* { **db-endpoint** { **host** *host_name* | **port** *port_number* }
| **kafka-server** { *host_name* *port_number* } }

system_id

リモートシステムまたはサイトの ID を指定します。

整数である必要があります。

使用上のガイドライン このコマンドを使用して、リモートサイトのリストを設定します。

cdl remote-site db-endpoint

リモート DB エンドポイントのリストを設定します。

コマンドモード [Exec] > [Global Configuration]

構文の説明 **cdl remote-site** *system_id* **db-endpoint** { **host** *host_name* | **port** *port_number* }

host *host_name*

リモート DB エンドポイントのホスト名または IP アドレスを指定します。

1 ～ 64 文字の文字列である必要があります。

port *port_number*

リモート DB エンドポイントのポート番号を指定します。

使用上のガイドライン このコマンドを使用して、リモート DB エンドポイントのリストを設定します。

cdl remote-site kafka-server

Kafka サーバーのリストを設定します。

コマンド モード

[Exec] > [Global Configuration]

構文の説明

```
cdl remote-site system_id kafka-server host_name port_number ssl-port
ssl_port_number
```

ssl-port *ssl_port_number*

リモート Kafka ブローカによって公開される SSL ポートを指定します。

host_name

リモート Kafka サーバーのホスト名または IP アドレスを指定します。

1 ～ 64 文字の文字列である必要があります。

port_number

リモート Kafka サーバーのポート番号を指定します。

使用上のガイドライン このコマンドを使用して、Kafka サーバーのリストを設定します。

cdl ssl-config

SSL 設定パラメータを設定します。

コマンド モード

[Exec] > [Global Configuration]

構文の説明

```
cdl ssl-config enable { false | true }
```

enable { false | true };

SSL 設定を有効にするか無効にするかを指定します。

次のいずれかである必要があります。

- false
- true

デフォルト値は false です。

使用上のガイドライン このコマンドを使用して、SSL 設定パラメータを設定します。

cdl ssl-config certs

データベースエンドポイント固有の SSL 設定パラメータを設定します。

コマンド モード [Exec] > [Global Configuration]

構文の説明 **certs ip** *fqdn_ip_address* **ssl-key** *ssl_key_file* **ssl-ca** *ssl_ca* **ssl-crt** *ssl_cert_file*

host *hostname_ip_address*

外部ホストのホスト名または IP アドレスを指定します。

文字列である必要があります。

ssl-ca *ssl_ca*

CA 証明書ファイルの内容を指定します。

ssl-crt *ssl_cert_file*

SSL 証明書ファイルを指定します。

ssl-key *ssl_key_file*

SSL キーファイルを指定します。

使用上のガイドライン このコマンドを使用して、データベースエンドポイント固有の SSL 設定パラメータを設定します。

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。