

CDL Backup and Restoration

This chapter describes the operational procedures for CDL backup and restoration.

- Access CDL Ops-Center Configuration, on page 1
- Backup and Restore, on page 2

Access CDL Ops-Center Configuration

Use this procedure to access the CDL API configuration.

Before you begin

Ensure that the CDL datastore is configured to configure CDL API.

Procedure

Step 1 Enable the CDL API pod deployment using the **cdl api deploy true** command.

By default, the pod deployment is disabled.

Step 2 Scale the CDL API pod vertically using the following commands.

a) Configure the maximum CPU limit using the **cdl api cpu-limit** *cpu_limit* command.

Example:

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

cpu_limit must be an integer in the range of 1000 to 8000. Default: 4000.

b) Configure the maximum number of CPU cores using the **cdl api go-max-procs** *cpu_cores* command.

Example:

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

cpu_cores must be an integer in the range of 1 to 32. Default: 4.

c) Configure the maximum memory allowed in Gi using the **cdl api memory-limit** *memory_limit* command.

Example:

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

memory_limit must be an integer in the range of 1 to 8. Default: 4.

d) Configure the backup storage in GB using the **cdl api storage** *storage_size* command.

Example:

```
cdl api storage 90
```

storage_size must be an integer in the range of 1 to 200. Default: 90.

Backup and Restore

Use this procedure to trigger and restore the backup process.

Before you begin

Ensure that you configure **cdl api deploy true** to access the **cdl actions backup** and **cdl actions restore** commands.

Procedure

Step 1 Trigger the backup process using the **cdl actions backup start [concurrency-factor** *concurrency_factor* **| created-before** *timestamp* **| slice-name** *slice_name* **]** command.

This command triggers the backup process and will not wait for the process to be completed. It returns a successful or failure message with the dump file name that gets created. Once the backup process is completed, the dump file in the output can be collected or referred for the restore process. This file can be located at */data/<namespace>/data-cdl-api-0/<file-name.dump>* on the node where the cdl-api-0 pod is running.

This command will prompt the warning *This triggers CDL backup process. Are you sure?*. Type *yes* to proceed with triggering the CDL backup process.

The following parameters are optional to be configured with the **backup start** command:

- **concurrency-factor** *concurrency_factor*—Take the backup by fetching data from the slots concurrently. By, default only 1 slot will be queried at a time.

  concurrency_factor must be an integer in the range 1 to 8.

- **created-before** *timestamp*—Take the backup of records created before this timestamp. The timestamp must be in the `CCYY-MM-DDTHH:MM:SS` format.

  By default, all available data will be queried for backup.

- **slice-name** *slice_name*—Take the backup of the given slice names. Only the configured slice name will be allowed to pass here.

  By default, the backup of all slice names will be taken.

Step 2 Run the **cdl actions backup status** command to know the status of the backup process.

This command returns one of the following status:

- **idle | completed**—CDL backup process is completed or not going on
- **in-progress**—CDL backup process is in-progress

Step 3 Trigger the restore process using the **cdl actions restore start file** *filename* command.

This command triggers the restore process and will not wait for the process to be completed. It returns a successful or failure message. It is mandatory to configure the **`file`** parameter with this command.

This command will prompt the warning *This triggers CDL restore process and will overwrite the existing data. Are you sure?*. Type *yes* to proceed with triggering the CDL restore process.

The file name must have the `.dump` extension. The file must be present in the host path */data/<namespace>/data-cdl-api-0/<file-name.dump>* on the node where the cdl-api-0 pod is running.

Step 4 Run the **cdl actions restore status** command to know the status of the restore process.

This command returns one of the following status:

- **idle | completed**—CDL restore process is completed or not going on
- **in-progress**—CDL restore process is in-progress