



Dremio Software

How to Migrate Catalog Objects From Dremio Software to Dremio Cloud

Introduction

The purpose of this document is to guide you through the steps for migrating your semantic layer from Software versions of Dremio, which could be Community Edition or Enterprise Edition installed on Kubernetes, AWSE or VMs, to Dremio Cloud, using Dremio Cloner to perform the read of resources from one environment and write into the other.

Audience

Dremio Cloner must be used by someone with access to Dremio Administrator credentials.

Limitations

The following limitations apply when using Dremio Cloner to migrate from Software to Dremio Cloud.

Loss of Resources in Home Spaces

One of the unavoidable consequences of migrating from Software to Dremio Cloud is that any resources created inside the user's home spaces will be lost as part of this process. Therefore please be sure to make copies of any uploaded files that users want to re-upload after the migration, or perhaps move them into data lake storage. Also, if there are any VDSs in home spaces that users want to keep, consider moving them into shared spaces before beginning the migration.

Loss of Datasets if Their Chain of Execution is Broken

If a VDS is not executable, for example if it makes use of VDSs/PDSs that have been removed, then the VDS will not be created in Dremio Cloud.

Loss of resources if some data sources are not put into Dremio Cloud

There may be some data sources that you do not want to migrate to Dremio Cloud or indeed cannot be created in Dremio Cloud. In these circumstances where data sources are omitted from Dremio Cloud then any PDSs associated with the data source will also not be created. Similarly, any VDSs that try to make use of PDSs in the missing data sources will also fail to get created in Dremio Cloud.

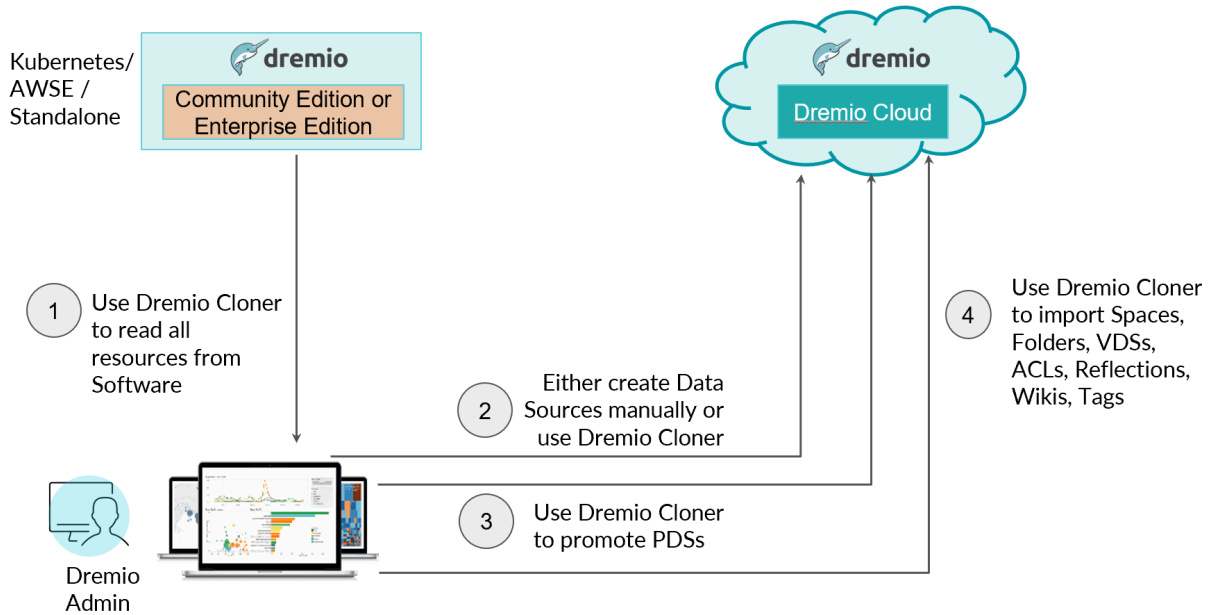
Recommend Recreating Data Sources Manually

When the semantic layer is being written into Dremio Cloud, to avoid the need to hard-code any data source passwords in clear text in the JSON file produced by Dremio Cloner during the [“get” from Software steps](#), Dremio Professional Services recommends first recreating all of the data sources manually via the Dremio Cloud UI before restoring any other objects. This will also help to confirm that they have been configured correctly.

All of the connection details for every source (except for the passwords) will be saved in the JSON file, which can assist in this exercise.

Migration Process

Below is a diagram of the high-level process we will follow during the Software to Dremio Cloud migration:



Prerequisites

Dremio Cloner can be installed on any Linux server that has access to the Dremio Software Coordinator's REST API endpoints (typically on port 9047), it is assumed that the Software environment and Dremio Cloud are fully operational during the migration.

Dremio Cloner requires Python version 3.5 or later and some additional Python libraries (requests, SSL).

Before running any scripts, we want to **capture evidence** in the form of screenshots and/or SQL queries of how many resources are present in each space and data source in our environment. This is so that after we've migrated our semantic layer into Dremio Cloud we can compare the restored counts with the original counts to ensure all resources have been migrated, and if they haven't then we can investigate why.

IMPORTANT: Gather Connection Details and Logins\Passwords for all Data Sources

For every data source that is connected to Dremio that requires a password\access key to connect, the password\access key must be known, otherwise it will not be possible to migrate the data source to Dremio Cloud as part of this exercise.

Check For Spaces in Resource Names

In the catalog, search for the space character to see if any resources are returned. Make a note of the resources with spaces in their names, since Dremio Cloner has been reported to have intermittent issues importing these resources. This evidence will help identify resources that may fail to migrate due to having a space in their name.

Put space character in here and see what results are displayed

Category	Item	Count
Spaces (4)	Application	13
	Business	2
	Product	1
	Security	2
Sources	Databases (1)	
	pg	7

Evidence of Number of Resources in Spaces

Run the following two queries in the Dremio Software and save the results of each query in a CSV file. This first query lists the count of all resources per space, which is equivalent to the values shown against each space in the Dremio UI.

```

SELECT
  a.spaceName,
  count(a.spaceName)
FROM
  (select
    substr(TABLE_SCHEMA, 0, CASE WHEN STRPOS(TABLE_SCHEMA, '.') < 1 THEN
    LENGTH(TABLE_SCHEMA) ELSE STRPOS(TABLE_SCHEMA, '.') - 1 END ) as SpaceName
  FROM INFORMATION_SCHEMA.VIEWS) a
GROUP BY a.spaceName
ORDER by a.spaceName ASC

```

This query lists the names of all resources in each space:

```

SELECT
  substr(TABLE_SCHEMA, 0, CASE WHEN STRPOS(TABLE_SCHEMA, '.') < 1 THEN
  LENGTH(TABLE_SCHEMA) ELSE STRPOS(TABLE_SCHEMA, '.') - 1 END ) as SpaceName,
  CONCAT(TABLE_SCHEMA, '.', TABLE_NAME) AS ResourceName
FROM
  INFORMATION_SCHEMA.VIEWS
WHERE SUBSTR(TABLE_SCHEMA, 0, 1) != '@' -- Ignore home space resources
ORDER BY SpaceName ASC

```

Evidence of Number of PDSs in Data Sources

Use the following query in the Dremio Software UI to report on the number of PDSs in each data source. Save the results as a CSV file.

```

SELECT
  a.sourceName,
  count(a.sourceName)
FROM
  (select
    substr(TABLE_SCHEMA, 0, CASE WHEN STRPOS(TABLE_SCHEMA, '.') < 1 THEN
    LENGTH(TABLE_SCHEMA) ELSE STRPOS(TABLE_SCHEMA, '.') - 1 END ) as sourceName
  FROM INFORMATION_SCHEMA."TABLES"
  WHERE TABLE_TYPE = 'TABLE') a
GROUP BY a.sourceName
ORDER by a.sourceName ASC

```

Evidence of Reflections

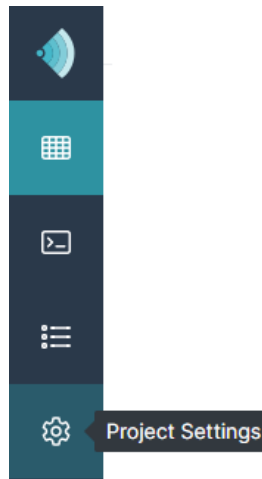
Run the following query in the Dremio Software UI to capture the names, associated datasets, and current status of all reflections set up in the environment. Save the results in a CSV file.

```
SELECT
  dataset,
  name,
  type,
  status
FROM
  sys.reflections
ORDER BY dataset ASC
```

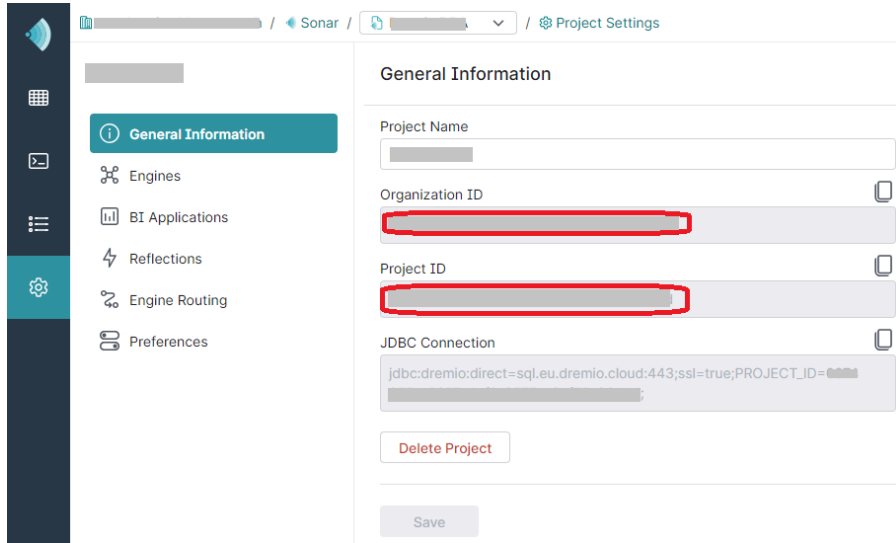
Gather Dremio Cloud Organization ID and Project ID

To migrate resources into Dremio Cloud, the Dremio Cloud Organization ID and Project ID for the Dremio Cloud instance you want to create resources into need to be set in the configuration files used by Dremio Cloner.

To gather these values, log into your respective Dremio Cloud project with an administrative user at either <https://app.dremio.cloud> or <https://app.eu.dremio.cloud>. Navigate to Project Settings on the left of the UI, as shown below:



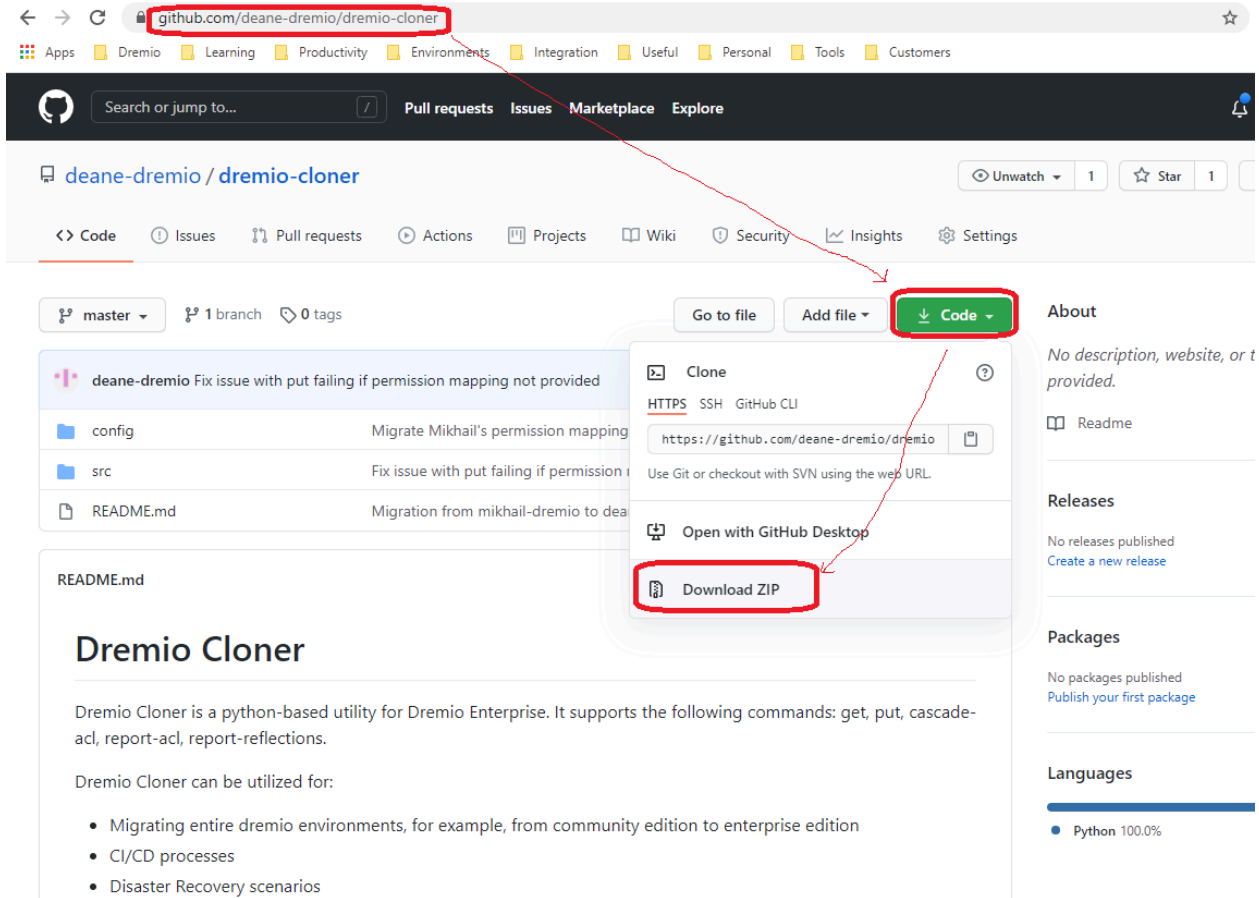
On the General Information page of the Project Settings, you will find both the Dremio Cloud Organization ID and Project ID listed. Take note of these values.



Install Dremio Cloner

The Dremio Cloner package can be downloaded from the following location:

<https://github.com/deane-dremio/dremio-cloner>



- Unzip the package onto a machine that can successfully make calls to the Dremio Software and Dremio Cloud REST API. You can run Dremio Cloner from either a Windows or Linux machine. In this document, we assume the tool is copied onto a Linux machine and unzipped into `/opt/tools/dremio-cloner`.

```
unzip dremio-cloner-master.zip -d /opt/tools
mv /opt/tools/dremio-cloner-master /opt/tools/dremio-cloner
```

- The tool also requires Python 3.5 or greater.

```
python -V
```

- If you are not presented with details of the currently installed Python version, then you will need to install Python. The following site provides an explanation if required:

<https://docs.python-guide.org/starting/installation/>

- Navigate to the `dremio-cloner` directory and issue the following command:

```
python dremio_cloner.py
```

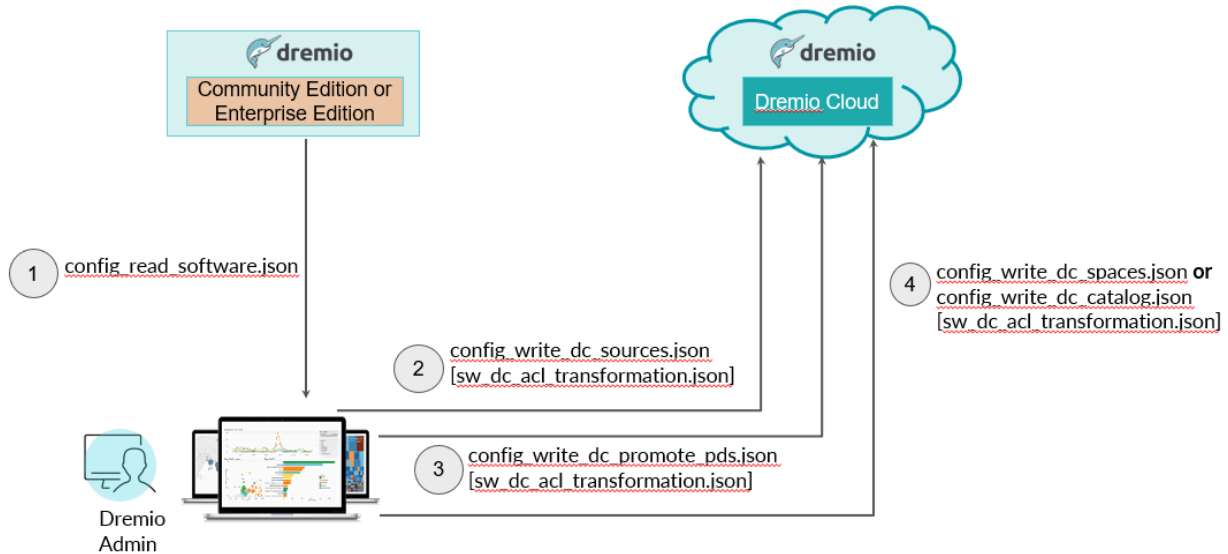
You should be presented with the following output which indicates the tool is installed and executable:

```
usage: dremio_cloner config_file -p password
Make sure the config file is correct.
```


Setup JSON configuration files

When it is executed, Dremio Cloner relies on a configuration file that tells it how to operate. As part of the use case to migrate from Software to Dremio Cloud, it requires that up to five distinct configuration files be created, one for exporting resources out of Dremio Software and up to four for importing resources into Dremio Cloud.

The following diagram shows which files are used during which parts of the migration process:



The following sections explain how to configure each of these files ready for use in the migration steps.

`config_read_software.json`

Appendix A of this document contains a JSON document with several placeholder values that need replacing with actual values.

The purpose of this JSON document is to specify exactly which resources we want to export out of the existing Dremio Software environment. This is used during step 1 outlined in the [migration process](#) section of this document.

These are the steps to create and configure the file:

- Copy the contents of Appendix A and write it to a file called `config_read_software.json`.
- The following table summarises the placeholder values to search for and a description of what the placeholder needs to change to.
- Once the placeholder values have all been replaced, copy the file into the `config` subfolder of the `dremio-cloner` tool, e.g. into `/opt/tools/dremio-cloner/config`.

Placeholder	Description
<DREMIO_UI_ENDPOINT>	The base URL for accessing the Dremio Software UI and REST API e.g. http://localhost:9047.
<DREMIO_ADMIN_USER>	Name of an administrative user in Dremio Software that can access the REST API
<OPTIONAL_PASSWORD>	<p>Password associated with an administrative user in Dremio Software that can access the REST API.</p> <p>You do have the option to delete the text <i><OPTIONAL_PASSWORD></i> and leave an empty string inside the set of double quotes, if you do this then you will be prompted to enter the password on the command line when running <code>dremio_cloner.py</code>.</p> <p>Alternatively, the password can be specified during the execution of <code>dremio_cloner.py</code> with a <code>-p</code> command line option.</p>
<TARGET_JSON_FILE_PATH_AND_NAME>	This tells Dremio Cloner where to write the objects exported out of Dremio Software as a JSON file. Needs to be a full path to a file, e.g.: <code>/opt/tools/dremio_cloner_sw_yyyymmdd.json</code> (where <code>yyymmdd</code> can be replaced with the current date information).
<LOG_FILE_NAME>	Dremio Cloner generates useful logging. This tells Dremio Cloner where to write the logs e.g. <code>/opt/tools/dremio_get.log</code>

Placeholders in `config_read_software.json` that need to be replaced with real values

sw_dc_acl_transformation.json

Appendix B contains a JSON document that can be used as a basis for specifying mappings between the users\groups that exist in Dremio Software and the equivalent users\groups that exist in Dremio Cloud (since the authentication mechanism and set of users/groups may be different between Software and Dremio Cloud). This mapping allows us to carry over as many ACLs as possible when the migration is performed, assuming the users/groups being mapped are already present in Dremio Cloud.

NOTE: If you do not want to carry over any ACLs during the migration and would prefer to set them up afterward then you can ignore setting up this file.

- Copy the contents of Appendix B and write it to a file called `sw_dc_acl_transformation.json`.
- Edit the file to create entries for all required mappings between Dremio Software usernames or group names onto equivalent usernames or group names in Dremio Cloud.
 - Example mapping an existing Dremio Authentication user to a user in Dremio Cloud:


```
{ "source": {"user": "mydremioauthuser1"},
  "target": {"user": "user@myorg.com"}
},
```
 - Example mapping an existing Dremio Authentication user to a group in Dremio Cloud:


```
{ "source": {"user": "mydremioauthuser2"},
  "target": {"group": "my-org-group"}
},
```
- Once all mappings have been created, copy the file into the `config` subfolder of the `dremio-cloner` tool, e.g. into `/opt/tools/dremio-cloner/config`.

config_write_dc_sources.json

Appendix C contains a JSON document with several placeholder values that need replacing with actual values.

This JSON document is optional in the migration process and is only required if you choose to deploy sources using Dremio Cloner rather than create them manually directly in Dremio Cloud.

The purpose of this JSON document is to tell Dremio Cloner to read the specified JSON file containing previously exported objects and import all sources into Dremio Cloud and in addition, where applicable, apply any ACLs set against the sources based upon the mappings specified in the `sw_dc_acl_transformation.json` file.

For this operation to succeed requires the JSON file containing the exported objects to be edited manually to supply passwords for all data sources directly in the file, which is often deemed a security risk, hence the recommendation to create the data sources manually directly in Dremio Cloud.

These are the steps to create and configure the file:

- Copy the contents of Appendix C and write it to a file called `config_write_dc_sources.json`.
- The following table summarises the placeholder values to search for and a description of what the placeholder needs to change to.
- Once the placeholder values have all been replaced, copy the file into the `config` subfolder of the `dremio-cloner` tool, e.g. into `/opt/tools/dremio-cloner/config`.

Placeholder	Description
<DREMIO_ADMIN_USER>	Name of an administrative user in Dremio Cloud that can access the REST API
<OPTIONAL_PASSWORD>	<p>Password associated with an administrative user in Dremio Cloud that can access the REST API.</p> <p>You do have the option to delete the text <code><OPTIONAL_PASSWORD></code> and leave an empty string inside the set of double quotes, if you do this then you will be prompted to enter the password on the command line when running <code>dremio_cloner.py</code>.</p> <p>Alternatively, the password can be specified during the execution of <code>dremio_cloner.py</code> with a <code>-p</code> command line option.</p>
<DREMIO_CLOUD_ORG_ID>	The Organization ID of your Organization in Dremio Cloud. See here for details on how to obtain this value.
<DREMIO_CLOUD_PROJECT_ID>	The Project ID of the project in Dremio Cloud where you want to migrate the sources into. See here for details on how to obtain this value.
<TARGET_JSON_FILE_PATH_AND_NAME>	This tells Dremio Cloner the location of the JSON file containing the objects we want to import into Dremio Cloud. Needs to be a full path to a file, e.g.: <code>/opt/tools/dremio_cloner_sw_yyyymmdd.json</code> (where <code>yyymmdd</code> can be replaced with the current date information).
<LOG_FILE_NAME>	Dremio Cloner generates useful logging. This tells Dremio Cloner where to write the logs e.g. <code>/opt/tools/dremio_put_pds.log</code>

Placeholders in `config_write_dc_sources.json` that need to be replaced with real values

- In addition to the placeholders, if there is no requirement to map ACLs between the Software and Dremio Cloud environments, then the following line can be **removed** from the `config_write_dc_sources.json` file:

```
{"transformation": {"acl": {"file": "../config/sw_dc_acl_transformation.json"}}},
```

config_write_dc_promote_pds.json

Appendix D contains a JSON document with several placeholder values that need replacing with actual values.

The purpose of this JSON document is to tell Dremio Cloner to promote files into PDSs for data lake sources, based on the PDS definitions in the JSON file exported previously by Dremio Cloner. In addition, where applicable, any ACLs set against the PDSs will also be applied based on the mappings specified in the `sw_dc_acl_transformation.json` file.

These are the steps to create and configure the file:

- Copy the contents of Appendix C and write it to a file called `config_write_dc_promote_pds.json`.
- The following table summarises the placeholder values to search for and a description of what the placeholder needs to change to.
- Once the placeholder values have all been replaced, copy the file into the `config` subfolder of the `dremio-cloner` tool, e.g. into `/opt/tools/dremio-cloner/config`.

Placeholder	Description
<DREMIO_ADMIN_USER>	Name of an administrative user in Dremio Cloud that can access the REST API
<OPTIONAL_PASSWORD>	<p>Password associated with an administrative user in Dremio Cloud that can access the REST API.</p> <p>You do have the option to delete the text <code><OPTIONAL_PASSWORD></code> and leave an empty string inside the set of double quotes, if you do this then you will be prompted to enter the password on the command line when running <code>dremio_cloner.py</code>.</p> <p>Alternatively, the password can be specified during the execution of <code>dremio_cloner.py</code> with a <code>-p</code> command line option.</p>
<DREMIO_CLOUD_ORG_ID>	The Organization ID of your Organization in Dremio Cloud. See here for details on how to obtain this value.
<DREMIO_CLOUD_PROJECT_ID>	The Project ID of the project in Dremio Cloud where you want to promote PDSs for data lake sources. See here for details on how to obtain this value.
<TARGET_JSON_FILE_PATH_AND_NAME>	<p>This tells Dremio Cloner the location of the JSON file containing the objects we want to promote in Dremio Cloud. Needs to be a full path to a file, e.g.:</p> <p><code>/opt/tools/dremio_cloner_ce_yyyymmdd.json</code> (where <code>yyymmdd</code> can be replaced with the current date information).</p>
<LOG_FILE_NAME>	Dremio Cloner generates useful logging. This tells Dremio Cloner where to write the logs e.g. <code>/opt/tools/dremio_put_pds.log</code>

Placeholders in `config_write_dc_promote_pds.json` that need to be replaced with real values

- In addition to the placeholders, if there is no requirement to map user/group ACLs between the Software and Dremio Cloud environments, then the following line can be **removed** from the `config_write_dc_promote_pds.json` file:

```
{"transformation": {"acl": {"file": "../config/sw_dc_acl_transformation.json"}}},
```

config_write_dc_spaces.json

NOTE: If ARS is disabled in your Dremio Cloud project please skip this section and refer to the next section titled [config_write_dc_catalog.json](#) instead.

Appendix E contains a JSON document with several placeholder values that need replacing with actual values.

The purpose of this JSON document is to tell Dremio Cloner to read the specified JSON file containing previously exported objects and import all spaces, folders, VDSs, reflection definitions, ACLs, wiki and tags into Dremio Cloud.

These are the steps to follow:

- Copy the contents of Appendix E and write it to a file called `config_write_dc_spaces.json`.
- The following table summarises the placeholder values to search for and a description of what the placeholder needs to change to.
- Once the placeholder values have all been replaced, copy the file into the config subfolder of the dremio-cloner tool, e.g. into `/opt/tools/dremio-cloner/config`.

Placeholder	Description
<DREMIO_ADMIN_USER>	Name of an administrative user in Dremio Cloud that can access the REST API
<OPTIONAL_PASSWORD>	<p>Password associated with an administrative user in Dremio Cloud that can access the REST API.</p> <p>You do have the option to delete the text <code><OPTIONAL_PASSWORD></code> and leave an empty string inside the set of double quotes, if you do this then you will be prompted to enter the password on the command line when running <code>dremio_cloner.py</code>.</p> <p>Alternatively, the password can be specified during the execution of <code>dremio_cloner.py</code> with a <code>-p</code> command line option.</p>
<DREMIO_CLOUD_ORG_ID>	The Organization ID of your Organization in Dremio Cloud. See here for details on how to obtain this value.
<DREMIO_CLOUD_PROJECT_ID>	The Project ID of the project in Dremio Cloud where you want to migrate resources. See here for details on how to obtain this value.

<TARGET_JSON_FILE_PATH_AND_NAME>	This tells Dremio Cloner the location of the JSON file containing the objects we want to import into Dremio Cloud. Needs to be a full path to a file, e.g.: /opt/tools/dremio_cloner_ce_yyyymmdd.json (where yyyymmdd can be replaced with the current date information).
<LOG_FILE_NAME>	Dremio Cloner generates useful logging. This tells Dremio Cloner where to write the logs e.g. /opt/tools/dremio_put_spaces.log

Placeholders in config_write_dc_spaces.json that need to be replaced with real values

- In addition to the placeholders, if there is no requirement to map user/group ACLs between the Software and Dremio Cloud environments, then the following line can be **removed** from the config_write_dc_spaces.json file:

```
{"transformation": {"acl": {"file": "../config/sw_dc_acl_transformation.json"}}},
```

config_write_dc_catalog.json

NOTE: If ARS is disabled in your Dremio Cloud project please skip this section. Refer to the previous section titled [config_write_dc_spaces.json](#) for details of non-ARS configuration.

Appendix F contains a JSON document with several placeholder values that need replacing with actual values.

The purpose of this JSON document is to tell Dremio Cloner to read the specified JSON file containing previously exported objects and import all spaces, folders, VDSs, reflection definitions, ACLs, wiki and tags into Dremio Cloud into the catalog named in the JSON file.

These are the steps to follow:

- Copy the contents of Appendix F and write it to a file called config_write_dc_cataog.json.
- The following table summarises the placeholder values to search for and a description of what the placeholder needs to change to.
- Once the placeholder values have all been replaced, copy the file into the config subfolder of the dremio-cloner tool, e.g. into /opt/tools/dremio-cloner/config.

Placeholder	Description
<DREMIO_ADMIN_USER>	Name of an administrative user in Dremio Cloud that can access the REST API

<OPTIONAL_PASSWORD>	<p>Password associated with an administrative user in Dremio Cloud that can access the REST API.</p> <p>You do have the option to delete the text <code><OPTIONAL_PASSWORD></code> and leave an empty string inside the set of double quotes, if you do this then you will be prompted to enter the password on the command line when running <code>dremio_cloner.py</code>.</p> <p>Alternatively, the password can be specified during the execution of <code>dremio_cloner.py</code> with a <code>-p</code> command line option.</p>
<DREMIO_CLOUD_ORG_ID>	The Organization ID of your Organization in Dremio Cloud. See here for details on how to obtain this value.
<DREMIO_CLOUD_PROJECT_ID>	The Project ID of the project in Dremio Cloud where you want to migrate resources. See here for details on how to obtain this value.
<DEFAULT-ARCTIC-CATALOG-NAME>	The name of the catalog in Dremio Cloud that you want to migrate resources into.
<TARGET_JSON_FILE_PATH_AND_NAME>	<p>This tells Dremio Cloner the location of the JSON file containing the objects we want to import into Dremio Cloud. Needs to be a full path to a file, e.g.:</p> <p><code>/opt/tools/dremio_cloner_ce_yyyymmdd.json</code> (where <code>yyymmdd</code> can be replaced with the current date information).</p>
<LOG_FILE_NAME>	<p>Dremio Cloner generates useful logging. This tells Dremio Cloner where to write the logs e.g.</p> <p><code>/opt/tools/dremio_put_spaces.log</code></p>

Placeholders in `config_write_dc_catalog.json` that need to be replaced with real values

- In addition to the placeholders, if there is no requirement to map user/group ACLs between the Software and Dremio Cloud environments, then the following line can be **removed** from the `config_write_dc_spaces.json` file:

```
{"transformation": {"acl": {"file": "../config/sw_dc_acl_transformation.json"}}},
```

Best Practices

Below is a list of best practices to consider when running through the migration steps detailed in the following sections of this document.

- Take a [backup](#) of the Dremio Software metadata database
- It is highly recommended that the target Dremio Cloud project is an empty project with no objects previously created within it. If the project has any pre-defined semantic layer whose resource names clash with those being migrated from Software then note that they will be overwritten/updated.

- Create data sources manually in the target Dremio Cloud project rather than rely on Dremio Cloner to create them, thus avoiding potential security concerns of needing data source passwords and access keys in clear text.
 - By design, the tool will not store the passwords or access keys on disk when it reads data source metadata out of Dremio Software.
 - If you want to leverage the Dremio Cloner to create the data sources you have to update the JSON file exported from Dremio Software to embed the passwords and/or access keys.
- Make a copy of the original template files in the Dremio Cloner package (perhaps copy `-r dremio-cloner-master/config` to `dremio-cloner-master/config_orig`)
- This document assumes we will be using Dremio Cloner in File Mode (where one big JSON file will be generated from the metadata exported from Dremio Software). However, for extremely large Software environments consider using Dremio Cloner in Directory Mode instead (where each object will be written to its own file on disk).
 - One benefit of Directory Mode is it can be easier to debug warnings or errors if you need to look at PDS / VDS definitions
 - It can also be easier to manipulate files to include passwords/access keys before writing to the target environment if you want to use Dremio Cloner to create the data sources
- It is **not recommended** to attempt changing space names or source names as part of the migration since this could involve heavy manual manipulation of the JSON file generated from Dremio Software and is highly error-prone.
- Note in the `config_write_dc_sources.json`, `config_write_dc_promote_pds.json` and `config_write_dc_spaces.json` files we specify `vds.max_hierarchy_depth": "25"`
 - This specifies the number of times the code will loop through the dependencies in the objects to ensure we have covered all objects. A hierarchy depth of 25 is assumed to be sufficient, rarely there is an environment with more than 10 levels.
- Review the log file for ERRORS with the GET/READ and the various PUT/WRITE operations.

Export Resources from Software

The first step of this migration as mentioned in the best practices above will be to perform a backup of the Dremio Software metadata database using the [dremio-admin backup](#) command.

Once this is complete we will export resources out of Dremio Software using Dremio Cloner and the [config_read_software.json](#) configuration file.

Dremio Cloner “get” from Software

The following steps assume you have created the configuration files detailed in earlier sections of this document.

- Navigate to the src folder of the dremio-cloner package

```
cd /opt/tools/dremio-cloner/src
```

- Execute the following Dremio Cloner command to export resources out of Dremio Software, this may take several minutes to complete depending on the number of objects that are being exported. There will be no feedback on the command line while this is in progress, however, the log file specified in the `config_read_software.json` configuration file will be getting filled with debug information. If you did not specify the password of the admin user in the configuration file then you will be prompted to enter it.

```
python dremio_cloner.py ../config/config_read_software.json
```

Alternatively, you can enter the password using an additional `-p` option, for example:

```
python dremio_cloner.py ../config/config_read_software.json -p adminpwd
```

This will result in a file being written to disk in the location specified by the **filename** element in the `config_read_software.json` configuration file if using the default File Mode. If using Directory Mode then this will result in multiple files being written to disk in the location specified by the **directory** element in the `config_read_software.json` configuration file.

Deploy Resources into Dremio Cloud

Data Sources

Create Data Sources Manually

To avoid the need to hard-code any data source passwords in clear text in the JSON file produced by Dremio Cloner during the export from Dremio Software, it is recommended to recreate all of the data sources manually via the Dremio UI. This will also help to confirm that they have been configured correctly.

All of the source names, connection details and configuration settings for every source (except for the passwords) will be saved in the JSON file that contains the metadata exported from Dremio Software, which can assist in this exercise.

Create Data Sources Using Dremio Cloner

If you choose to hard-code the data source passwords in clear text in the JSON file produced by Dremio Cloner then the following steps can be used to create the sources in Dremio Cloud using Dremio Cloner.

- Navigate to the src folder of the dremio-cloner package

```
cd /opt/tools/dremio-cloner/src
```

- Execute the following Dremio Cloner command, this may take several minutes to complete depending on the number of objects that are being restored. There will be no feedback on the command line while this is in progress, however, the log file specified in the `config_write_dc_sources.json` configuration file will be getting filled with debug information. If you did not specify the password of the admin user in the configuration file then you will be prompted to enter it.

```
python dremio_cloner.py ../config/config_write_dc_sources.json
```

Alternatively, you can enter the password using an additional `-p` option, for example:

```
python dremio_cloner.py ../config/config_write_dc_sources.json -p adminpwd
```

- Check the Dremio Cloner log files for any errors in creating the sources. Most likely failure reasons are misconfigured login credentials and failure for Dremio Cloud to access the data source due to network issues (perhaps a firewall hasn't been opened for example).

- If a source fails to get created using Dremio Cloner then try to create it manually via the Dremio UI instead and use Dremio Cloud diagnostics to help diagnose any further issues.

Promote Physical Datasets

Once the data sources are created, Dremio Cloner can be used to promote any PDSs for file-based data sources such as data lakes. The JSON file produced during the Dremio Software export steps contains the relevant details, therefore the following steps can be carried out to promote the PDS definitions:

- Navigate to the src folder of the dremio-cloner package

```
cd /opt/tools/dremio-cloner/src
```

- Execute the following Dremio Cloner command, this may take several minutes to complete depending on the number of objects that are being restored. There will be no feedback on the command line while this is in progress, however, the log file specified in the `config_write_dc_promote_pds.json` configuration file will be getting filled with debug information. If you did not specify the password of the admin user in the configuration file then you will be prompted to enter it.

```
python dremio_cloner.py ../config/config_write_dc_promote_pds.json
```

Alternatively, you can enter the password using an additional `-p` option, for example:

```
python dremio_cloner.py ../config/config_write_dc_promote_pds.json -p adminpwd
```

- Compare the number of PDSs associated with each data source in the Dremio UI with the number of PDSs associated with each data source captured during the [prerequisites](#), these should match. If they do not match, check the Dremio Cloner log file for errors indicating why certain PDSs may not have been created. Below is an example error message indicating a reason why a PDS failed to get promoted, in this case, it is because the definition in the JSON file exported by the Dremio Cloner backup does not match the current structure of the file in the data source:

```
ERROR:2020-11-23 14:43:09,809:create_catalog_entity: received HTTP Response  
Code 400 for : <api/v3/catalog/> errorMessage: Column 'queryText' not found in  
any table  
  
ERROR:2020-11-23 14:43:09,809:_write_entity: could not create entity:  
dataset:HealthCheck/Preparation/results
```

If you encounter the same or similar errors in the logs then in the Dremio UI attempt to manually promote the offending files to PDSs.

Create Spaces and Associated Objects

Once the PDSs are promoted, Dremio Cloner can be used to create all objects associated with Spaces (e.g. Spaces, Folders, VDSs, Reflections, Wikis, Tags) in addition to Reflection definitions. However, the options that you supply into the configuration file for Dremio Cloner will differ depending on whether the Dremio Cloud project you are deploying these resources into has Arctic Replacing Spaces (ARS) enabled or not.

If ARS is not enabled then the creation of spaces and associated objects follows a very similar pattern to Dremio Software, where the top-level artifact in your resource tree is a Space. However, with ARS enabled this introduces an extra level of depth to the hierarchy of your catalog objects; the top-level artifact in your resource tree is no longer a Space, instead it is an Arctic catalog. So where in Dremio Software you may have a resource whose fully qualified name is `myspace.myfolder.myview`, in Dremio Cloud the fully qualified name of that same resource will be `mycatalog.myspace.myfolder.myview`. This needs to be taken into consideration during the migration, specifically in the Dremio Cloner configuration file, since every object being migrated will need to add a reference to the name of the catalog in its fully qualified name.

Dremio Cloud Projects with ARS Disabled

NOTE: If ARS is enabled in your Dremio Cloud project please skip this section and read the next one instead.

The JSON file produced during the backup steps contains the relevant details, therefore the following steps can be carried out to recover these definitions:

- Navigate to the `src` folder of the `dremio-cloner` package

```
cd /opt/tools/dremio-cloner/src
```

- Execute the following Dremio Cloner command, this may take several minutes to complete depending on the number of objects that are being restored. There will be no feedback on the command line while this is in progress, however, the log file specified in the `config_write_dc_spaces.json` configuration file will be getting filled with debug information. If you did not specify the password of the admin user in the configuration file then you will be prompted to enter it.

```
python dremio_cloner.py ../config/config_write_dc_spaces.json
```

Alternatively, you can enter the password using an additional `-p` option, for example:

```
python dremio_cloner.py ../config/config_write_dc_spaces.json -p adminpwd
```

- Compare the number of VDSs associated with each Space in the Dremio UI with the number of VDSs associated with each Space captured during the [prerequisites](#), these should match. If they do not match, check the Dremio Cloner log file for errors indicating why certain VDSs may not have been created.
 - If a VDS relies on a parent VDS or PDS to be created before it can be created and the parent isn't present, then the VDS will fail to get created.
 - With chains of VDSs, the failure of one VDS to get created can cause any subsequent VDSs that rely on that VDS to also fail to be created.

Dremio Cloud Projects with ARS Enabled

NOTE: If ARS is disabled in your Dremio Cloud project please skip this section and refer to the previous one instead.

The JSON file produced during the backup steps contains the relevant details, therefore the following steps can be carried out to recover these definitions:

- Navigate to the `src` folder of the `dremio-cloner` package

```
cd /opt/tools/dremio-cloner/src
```

- Execute the following Dremio Cloner command, this may take several minutes to complete depending on the number of objects that are being restored. There will be no feedback on the command line while this is in progress, however, the log file specified in the `config_write_dc_catalog.json` configuration file will be getting filled with debug information. If you did not specify the password of the admin user in the configuration file then you will be prompted to enter it.

```
python dremio_cloner.py ../config/config_write_dc_catalog.json
```

Alternatively, you can enter the password using an additional `-p` option, for example:

```
python dremio_cloner.py ../config/config_write_dc_catalog.json -p adminpwd
```

- The command will create Folders under the catalog specified in the `dremio_cloud_target_catalog_name` parameter of the `config_write_dc_arctic.json` file to represent the Spaces that existed in the original Dremio Software environment.
- Compare the number of VDSs associated with each Space in the Dremio UI with the number of VDSs associated with each equivalent Folder inside the Catalog captured during the [prerequisites](#), these should match. If they do not match, check the Dremio Cloner log file for errors indicating why certain VDSs may not have been created.
 - If a VDS relies on a parent VDS or PDS to be created before it can be created and the parent isn't present, then the VDS will fail to get created.
 - With chains of VDSs, the failure of one VDS to get created can cause any subsequent VDSs that rely on that VDS to also fail to be created.

Getting Support for Dremio Cloner

Dremio Cloner is developed by Dremio Professional Services and is not covered by Dremio Support or its associated support policies. If you are having trouble leveraging Dremio Cloner please feel free to submit issues to the GitHub repository or purchase Dremio Professional Services. No SLAs are guaranteed for fixing of any issues or enhancements to the tool raised via GitHub. Dremio Cloner is provided free of charge for those users who wish to migrate resources on their own without support.

Appendix A - config_read_software.json

```

{"dremio_cloner": [
  {"command": "get"},
  {"source": [
    {"endpoint": "<DREMIO_UI_ENDPOINT>"},
    {"username": "<DREMIO_ADMIN_USER>"},
    {"password": "<OPTIONAL_PASSWORD>"},
    {"verify_ssl": "False"},
    {"is_community_edition": "True"},
    {"graph_api_support": "True"}]
  },
  {"target": [
    {"filename": "<TARGET_JSON_FILE_PATH_AND_NAME>"},
    {"overwrite": "True"}]
  },
  {"options": [
    {"logging.level": "logging.DEBUG"},
    {"logging.format": "%(levelname)s:%(asctime)s:%(message)s"},
    {"logging.filename": "<LOG_FILE_NAME>"},
    {"logging.verbose": "False"},

    {"max_errors": "9999"},
    {"http_timeout": "10"},

    {"user.process_mode": "process"},
    {"group.process_mode": "process"},
    {"space.process_mode": "process"},
    {"source.process_mode": "process"},
    {"reflection.process_mode": "process"},
    {"wlm.queue.process_mode": "process"},
    {"wlm.rule.process_mode": "process"},
    {"wiki.process_mode": "process"},
    {"tag.process_mode": "process"},
    {"home.process_mode": "skip"},
    {"vote.process_mode": "skip"},
    {"folder.process_mode": "process"},
    {"vds.process_mode": "process"},
    {"pds.process_mode": "process"},

    {"space.filter": "*"},
    {"space.exclude.filter": ""},
    {"space.folder.filter": "*"},
    {"space.folder.exclude.filter": ""},

    {"source.filter": "*"},
    {"source.exclude.filter": ""},
    {"source.folder.filter": "*"},
    {"source.folder.exclude.filter": ""},

    {"pds.filter": "*"},
    {"pds.exclude.filter": ""},
    {"pds.list.useapi": "False"},

    {"vds.filter": "*"},

```



```
    {"vds.exclude.filter":""},  
    {"vds.dependencies.process_mode":"skip"}]  
  ]]  
}
```

Appendix B - sw_dc_acl_transformation.json

```
{"acl-transformation":  
[  
  { "source": {"user": "<USER_1>"},  
    "target": {"user": "<USER_1>"}  
  },  
  { "source": {"user": "<USER_2>"},  
    "target": {"group": "<GROUP_2>"}  
  }  
]  
}
```

Appendix C - config_write_dc_sources.json

```

{"dremio_cloner": [
  {"command": "put"},
  {"target": [
    {"endpoint": "https://api.dremio.cloud/"},
    {"username": "<DREMIO_ADMIN_USER>"},
    {"password": "<OPTIONAL_PASSWORD>"},
    {"verify_ssl": "False"},
    {"is_community_edition": "False"},
    {"is_dremio_cloud": "True"},
    {"dremio_cloud_org_id": "<DREMIO_CLOUD_ORG_ID>"},
    {"dremio_cloud_project_id": "<DREMIO_CLOUD_PROJECT_ID>"}
  ]
},
{"source": [
  {"filename": "<TARGET_JSON_FILE_PATH_AND_NAME>"}
],
{"options": [

  {"dry_run": "False"},

  {"logging.level": "logging.DEBUG"},
  {"logging.format": "%(levelname)s:%(asctime)s:%(message)s"},
  {"logging.filename": "<LOG_FILE_NAME>"},
  {"logging.verbose": "False"},

  {"max_errors": "9999"},
  {"http_timeout": "10"},
  {"source.retry_timedout": "false"},

  {"space.process_mode": "skip"},
  {"folder.process_mode": "skip"},
  {"source.process_mode": "create_overwrite"},
  {"pds.process_mode": "skip"},
  {"vds.process_mode": "skip"},
  {"reflection.process_mode": "skip"},
  {"pds.reflection_refresh_mode": "skip"},
  {"wiki.process_mode": "skip"},
  {"tag.process_mode": "skip"},
  {"home.process_mode": "skip"},
  {"vote.process_mode": "skip"},
  {"wlm.queue.process_mode": "skip"},
  {"wlm.rule.process_mode": "skip"},

  {"transformation": {"acl": {"file": "../config/sw_dc_acl_transformation.json"}}},

  {"space.ignore_missing_acl_user": "False"},
  {"space.ignore_missing_acl_group": "False"},
  {"folder.ignore_missing_acl_user": "False"},
  {"folder.ignore_missing_acl_group": "False"},
  {"source.ignore_missing_acl_user": "False"},
  {"source.ignore_missing_acl_group": "False"},
  {"pds.ignore_missing_acl_user": "False"},
  {"pds.ignore_missing_acl_group": "False"},

```

```
    {"vds.ignore_missing_acl_user": "False"},
    {"vds.ignore_missing_acl_group": "False"},

    {"space.filter": "*"},
    {"space.exclude.filter": ""},
    {"space.folder.filter": "*"},
    {"space.folder.exclude.filter": ""},

    {"source.filter": "*"},
    {"source.exclude.filter": ""},
    {"source.folder.filter": "*"},
    {"source.folder.exclude.filter": ""},

    {"reflection.filter_mode": "apply_vds_pds_filter"},

    {"pds.filter": "*"},
    {"pds.exclude.filter": ""},
    {"pds.list.useapi": "False"},

    {"vds.filter": "*"},
    {"vds.exclude.filter": ""},
    {"vds.max_hierarchy_depth": "25"}]
  }
}
```

Appendix D - config_write_dc_promote_pds.json

```

{"dremio_cloner": [
  {"command": "put"},
  {"target": [
    {"endpoint": "https://api.dremio.cloud/"},
    {"username": "<DREMIO_ADMIN_USER>"},
    {"password": "<OPTIONAL_PASSWORD>"},
    {"verify_ssl": "False"},
    {"is_community_edition": "False"},
    {"is_dremio_cloud": "True"},
    {"dremio_cloud_org_id": "<DREMIO_CLOUD_ORG_ID>"},
    {"dremio_cloud_project_id": "<DREMIO_CLOUD_PROJECT_ID>"}]
  },
  {"source": [
    {"filename": "<TARGET_JSON_FILE_PATH_AND_NAME>"}]
  },
  {"options": [

    {"dry_run": "False"},

    {"logging.level": "logging.DEBUG"},
    {"logging.format": "%(levelname)s:%(asctime)s:%(message)s"},
    {"logging.filename": "<LOG_FILE_NAME>"},
    {"logging.verbose": "False"},

    {"max_errors": "9999"},
    {"http_timeout": "10"},
    {"source.retry_timedout": "false"},

    {"space.process_mode": "skip"},
    {"folder.process_mode": "skip"},
    {"source.process_mode": "skip"},
    {"pds.process_mode": "promote"},
    {"vds.process_mode": "skip"},
    {"reflection.process_mode": "skip"},
    {"pds.reflection_refresh_mode": "skip"},
    {"wiki.process_mode": "skip"},
    {"tag.process_mode": "skip"},
    {"home.process_mode": "skip"},
    {"vote.process_mode": "skip"},
    {"wlm.queue.process_mode": "skip"},
    {"wlm.rule.process_mode": "skip"},

    {"transformation": {"acl": {"file": "../config/sw_dc_acl_transformation.json"}}},

    {"space.ignore_missing_acl_user": "False"},
    {"space.ignore_missing_acl_group": "False"},
    {"folder.ignore_missing_acl_user": "False"},
    {"folder.ignore_missing_acl_group": "False"},
    {"source.ignore_missing_acl_user": "False"},
    {"source.ignore_missing_acl_group": "False"},
    {"pds.ignore_missing_acl_user": "False"},
    {"pds.ignore_missing_acl_group": "False"},
    {"vds.ignore_missing_acl_user": "False"},
  ]}

```

```
{"vds.ignore_missing_acl_group":"False"},

{"space.filter": "*"},
{"space.exclude.filter": ""},
{"space.folder.filter":"*"},
{"space.folder.exclude.filter":""},

{"source.filter":"*"},
{"source.exclude.filter":""},
{"source.folder.filter":"*"},
{"source.folder.exclude.filter":""},

{"reflection.filter_mode": "apply_vds_pds_filter"},

{"pds.filter":"*"},
{"pds.exclude.filter":""},
{"pds.list.useapi":"False"},

{"vds.filter":"*"},
{"vds.exclude.filter":""},
{"vds.max_hierarchy_depth":"25"}]
}
```

Appendix E - config_write_dc_spaces.json

```

{"dremio_cloner": [
  {"command": "put"},
  {"target": [
    {"endpoint": "https://api.dremio.cloud/"},
    {"username": "<DREMIO_ADMIN_USER>"},
    {"password": "<OPTIONAL_PASSWORD>"},
    {"verify_ssl": "False"},
    {"is_community_edition": "False"},
    {"is_dremio_cloud": "True"},
    {"dremio_cloud_org_id": "<DREMIO_CLOUD_ORG_ID>"},
    {"dremio_cloud_project_id": "<DREMIO_CLOUD_PROJECT_ID>"},
    {"dremio_cloud_target_catalog_name": ""},
    {"dremio_cloud_spaces_to_catalog": "False"}]
  },
{"source": [
  {"filename": "<TARGET_JSON_FILE_PATH_AND_NAME>"}]
},
{"options": [

  {"dry_run": "False"},

  {"logging.level": "logging.DEBUG"},
  {"logging.format": "%(levelname)s:%(asctime)s:%(message)s"},
  {"logging.filename": "<LOG_FILE_NAME>"},
  {"logging.verbose": "False"},

  {"max_errors": "9999"},
  {"http_timeout": "10"},
  {"source.retry_timedout": "false"},

  {"space.process_mode": "create_overwrite"},
  {"folder.process_mode": "create_overwrite"},
  {"source.process_mode": "update_only"},
  {"pds.process_mode": "promote"},
  {"vds.process_mode": "create_overwrite"},
  {"reflection.process_mode": "create_overwrite"},
  {"pds.reflection_refresh_mode": "refresh"},
  {"wiki.process_mode": "create_overwrite"},
  {"tag.process_mode": "create_overwrite"},
  {"home.process_mode": "skip"},
  {"vote.process_mode": "skip"},
  {"wlm.queue.process_mode": "skip"},
  {"wlm.rule.process_mode": "skip"},

  {"transformation": {"acl": {"file": "../config/sw_dc_acl_transformation.json"}}},

  {"space.ignore_missing_acl_user": "False"},
  {"space.ignore_missing_acl_group": "False"},
  {"folder.ignore_missing_acl_user": "False"},
  {"folder.ignore_missing_acl_group": "False"},
  {"source.ignore_missing_acl_user": "False"},
  {"source.ignore_missing_acl_group": "False"},
  {"pds.ignore_missing_acl_user": "False"},

```

```
    {"pds.ignore_missing_acl_group":"False"},
    {"vds.ignore_missing_acl_user":"False"},
    {"vds.ignore_missing_acl_group":"False"},

    {"space.filter": "*"},
    {"space.exclude.filter": ""},
    {"space.folder.filter":"*"},
    {"space.folder.exclude.filter":""},

    {"source.filter":"*"},
    {"source.exclude.filter":""},
    {"source.folder.filter":"*"},
    {"source.folder.exclude.filter":""},

    {"reflection.filter_mode": "apply_vds_pds_filter"},

    {"pds.filter":"*"},
    {"pds.exclude.filter":""},
    {"pds.list.useapi":"False"},

    {"vds.filter":"*"},
    {"vds.exclude.filter":""},
    {"vds.max_hierarchy_depth":"25"}]
  }
}
```


Appendix F - config_write_dc_catalog.json

```

{"dremio_cloner": [
  {"command": "put"},
  {"target": [
    {"endpoint": "https://api.dremio.cloud/"},
    {"username": "<DREMIO_ADMIN_USER>"},
    {"password": "<OPTIONAL_PASSWORD>"},
    {"verify_ssl": "False"},
    {"is_community_edition": "False"},
    {"is_dremio_cloud": "True"},
    {"dremio_cloud_org_id": "<DREMIO_CLOUD_ORG_ID>"},
    {"dremio_cloud_project_id": "<DREMIO_CLOUD_PROJECT_ID>"},
    {"dremio_cloud_target_catalog_name": "<DEFAULT-ARCTIC-CATALOG-NAME>"},
    {"dremio_cloud_spaces_to_catalog": "True"}]
  },
{"source": [
  {"filename": "<TARGET_JSON_FILE_PATH_AND_NAME>"}]
},
{"options": [

  {"dry_run": "False"},

  {"logging.level": "logging.DEBUG"},
  {"logging.format": "%(levelname)s:%(asctime)s:%(message)s"},
  {"logging.filename": "<LOG_FILE_NAME>"},
  {"logging.verbose": "False"},

  {"max_errors": "9999"},
  {"http_timeout": "10"},
  {"source.retry_timedout": "false"},

  {"space.process_mode": "create_overwrite"},
  {"folder.process_mode": "create_overwrite"},
  {"source.process_mode": "update_only"},
  {"pds.process_mode": "promote"},
  {"vds.process_mode": "create_overwrite"},
  {"reflection.process_mode": "create_overwrite"},
  {"pds.reflection_refresh_mode": "refresh"},
  {"wiki.process_mode": "create_overwrite"},
  {"tag.process_mode": "create_overwrite"},
  {"home.process_mode": "skip"},
  {"vote.process_mode": "skip"},
  {"wlm.queue.process_mode": "skip"},
  {"wlm.rule.process_mode": "skip"},

  {"transformation": {"acl": {"file": "../config/sw_dc_acl_transformation.json"}}},

  {"space.ignore_missing_acl_user": "False"},
  {"space.ignore_missing_acl_group": "False"},
  {"folder.ignore_missing_acl_user": "False"},
  {"folder.ignore_missing_acl_group": "False"},
  {"source.ignore_missing_acl_user": "False"},
  {"source.ignore_missing_acl_group": "False"},
  {"pds.ignore_missing_acl_user": "False"},

```

```
    {"pds.ignore_missing_acl_group":"False"},
    {"vds.ignore_missing_acl_user":"False"},
    {"vds.ignore_missing_acl_group":"False"},

    {"space.filter": "*"},
    {"space.exclude.filter": ""},
    {"space.folder.filter":"*"},
    {"space.folder.exclude.filter":""},

    {"source.filter":"*"},
    {"source.exclude.filter":""},
    {"source.folder.filter":"*"},
    {"source.folder.exclude.filter":""},

    {"reflection.filter_mode": "apply_vds_pds_filter"},

    {"pds.filter":"*"},
    {"pds.exclude.filter":""},
    {"pds.list.useapi":"False"},

    {"vds.filter":"*"},
    {"vds.exclude.filter":""},
    {"vds.max_hierarchy_depth":"25"}]
  }
}
```