HCL MT CH-MSP Product Documentation

Importing Data

Planning to import data

Important: Please make sure that users from an organization do NOT login and access Connections BEFORE the data from the organization are imported!

- Process Flow for importing data
- The need for database import staging
- Setting up the S3 bucket
- TXT config file and Rclone parameters
- How to set an org to Read-Only

Process flow for importing data

Important: Please make sure that users from an organization do NOT login and access Connections BEFORE the data from the organization are imported!

Workflow for the initial request for binary files and DB Export files:

This first delivery of files will be used to copy the binary files into a staging environment, perform a DB import into staging environment and do the initial copy of binary files into the production environment.

1. **Customer** Selects CH-MSP and signs contract
2. **CH-MSP** creates a unique user ID in ServiceNow to manage the customer's account, do as soon as contract is in place. The CH-MSP needs to have a unique ID for each customer they are offboarding.
3. **Customer** adds CH-MSP user ID in ServiceNow to the customer accounts in ServiceNow. The CH-MSP will be added with regular user privileges.
4. **CH-MSP** opens a support case requesting to start the file transfer of the customers data. The orgid, customer name and CH-MSP email to invite for access to box need to be included in the case. Specify that this is the Initial data request for this org. Do NOT put S3 bucket information into the case. Support Template
5. **CH-MSP** add **scott.favreau@hcl.com** to the watch list for the case.
6. **HCL L2** creates a JIRA ticket for IBM to create a box folder to use for the offboarding activities for this customer. JIRA ticket includes orgid, customer name and CH-MSP email.
7. **HCL L2** creates a JIRA ticket for IBM to enable the customizer tool for the specified orgid.
8. **IBM** creates the box folder, invites the MSP to join, closes the JIRA ticket.
9. **HCL L2** to follow-up with MSP to make sure they received invitation to Box folder and tell them to put S3 information in box folder. A unique S3 Folder needs to be created for every customer that will be off boarded.
10. **IBM** enables the customizer tool for the Org and closes JIRA ticket.
11. **CH-MSP** puts S3 Bucket access information and security keys into Box folder and contacts HCL L2 support alerting team that S3 access information is available in Box and to start the File transfer process. The template must be used to provide S3 access information.
12. **HCL L2** creates JIRA ticket for IBM alerting them S3 information is in box folder and to start file transfer process.
13. **IBM** runs scripts to place Files in S3 bucket specified in box folder.
14. **IBM** runs DB export scripts to place database data in S3 bucket specified in box folder.
15. **IBM** closes JIRA ticket indicating that files have been copied to S3 bucket.
16. **IBM** deletes the box folder for this customer. Note, IBM records the S3 access information and will use this information during the second download of files.
17. **HCL Support** informs CH-MSP that the file transfer process has completed, and the files are in the S3 bucket.
18. **CH-MSP** copies files from S3 to their MT Connections Cloud.
19. **CH-MSP** copies binary files to appropriate location in MT Connections cloud (staging).
20. **CH-MSP** run the Import scripts to import data in the MT Connections DB (staging).
21. CH-MSP copies binary files to appropriate location in MT Connections cloud (production).
22. **CH-MSP and Customer** verify staging environment.
23. **MSP** confirms file transfer and HCL closes ticket.

Workflow for the request for the binary file update and DB Export files:

The second delivery of files will be used to update the binary files in the production environment (only files that have changed since initial download will be provided), perform a DB import into production environment. Note, the tar files that are downloaded in this step will overwrite the tar files that were loaded in the first request.

1. **MSP/Customer** determine when final cutover will occur
2. **Customer Admin** uses the customizer tool to mark the organization as read-only.
3. **CH-MSP** opens a support case with HCL requesting updated file transfer and DB export for customer org. The support case should specify Orgid/Customer name. Please specify this as a "Delta" data request.
   Support Template
4. **CH-MSP** add **scott.favreau@hcl.com** to the watch list for the case.

5. **HCL Support** opens a JIRA ticket for IBM requesting the File transfer and DB export be scheduled for the customer org specified in the ticket.
6. **IBM** runs scripts to place Files in S3 bucket specified in box folder. Only files modified/added since initial run will be placed in S3 bucket (this is handled by the scripts).
7. **IBM** runs export scripts and copies files to S3 bucket.
8. **IBM** closes JIRA ticket indicating that files have been copied to S3 bucket.
9. **HCL Support** informs CH-MSP that files are available.
10. **CH-MSP** copies files from S3 to their MT Connections Cloud.
11. **CH-MSP** copies files to appropriate location in MT Connections cloud (production).
12. **CH-MSP** imports the delta export data files to DB2 staging server
13. **CH-MSP** executes the external reference fix up scripts
14. **CH-MSP** run the Import scripts to import data from step 13 in the MT Connections DB (production).
15. **CH-MSP** Fixes any links, any other clean-up tasks.
16. **CH-MSP and Customer** verify data in MT Connections cloud.
17. **CH-MSP/Customer** confirms data is OK closes case.

The need for database import staging

This page explains the rationale behind a preliminary import of the databases and summarizes the actions you will need to take.

Converting external and guest users

Because the collaboration model for accommodating users external to your organization differs from the IBM Connections Cloud model, External users' identities and references to their identities need to be altered to accurately reflect them and the attribution of their content contributions. To do this a number of steps need to be performed on the data before it is imported into your production environment. These steps are crucial to ensure that external users are safely and accurately represented in your environment. There are scripts provided to perform the task, prepare the environment, do a preliminary import, perform data manipulation, and generate a set of exports that can be imported into your production environment. These scripts are found in the externalUsers.zip and the importUtils.zip script packages.

DB2 staging environment

To perform the conversion process, you will need to set up a DB2 server where the imported data can be stored temporarily while it is "massaged." This database server must have the Connections databases installed. For each org that you are preparing for import to production, you will need to do a preliminary import into this staging environment. Prior to performing the preliminary import, for
each org, you will need to ensure that all databases are clear of any residual organization data prior to the conversion of the data. Once the database is clean, you will run two scripts to prepare the databases to do the data manipulation.

Summary of database migration tasks

Here is the high-level list of tasks you will need to do for database migration:

- Prepare a staging server with "empty" Connections databases
- On staging server:
    - Run script to import data exported from IBM
    - Run script to fix-up external users and the content references
    - Run script to export the data
    - Run script to generate an ldif file for internal users
    - Run script to generate a CSV file for external users
- On Production server:
    - Run script to import massaged data to production server
    - Run script to import internal users to LDAP
- Start the process to re-register/re-invite external users

For details, see Importing the databases.

Setting up an S3 bucket

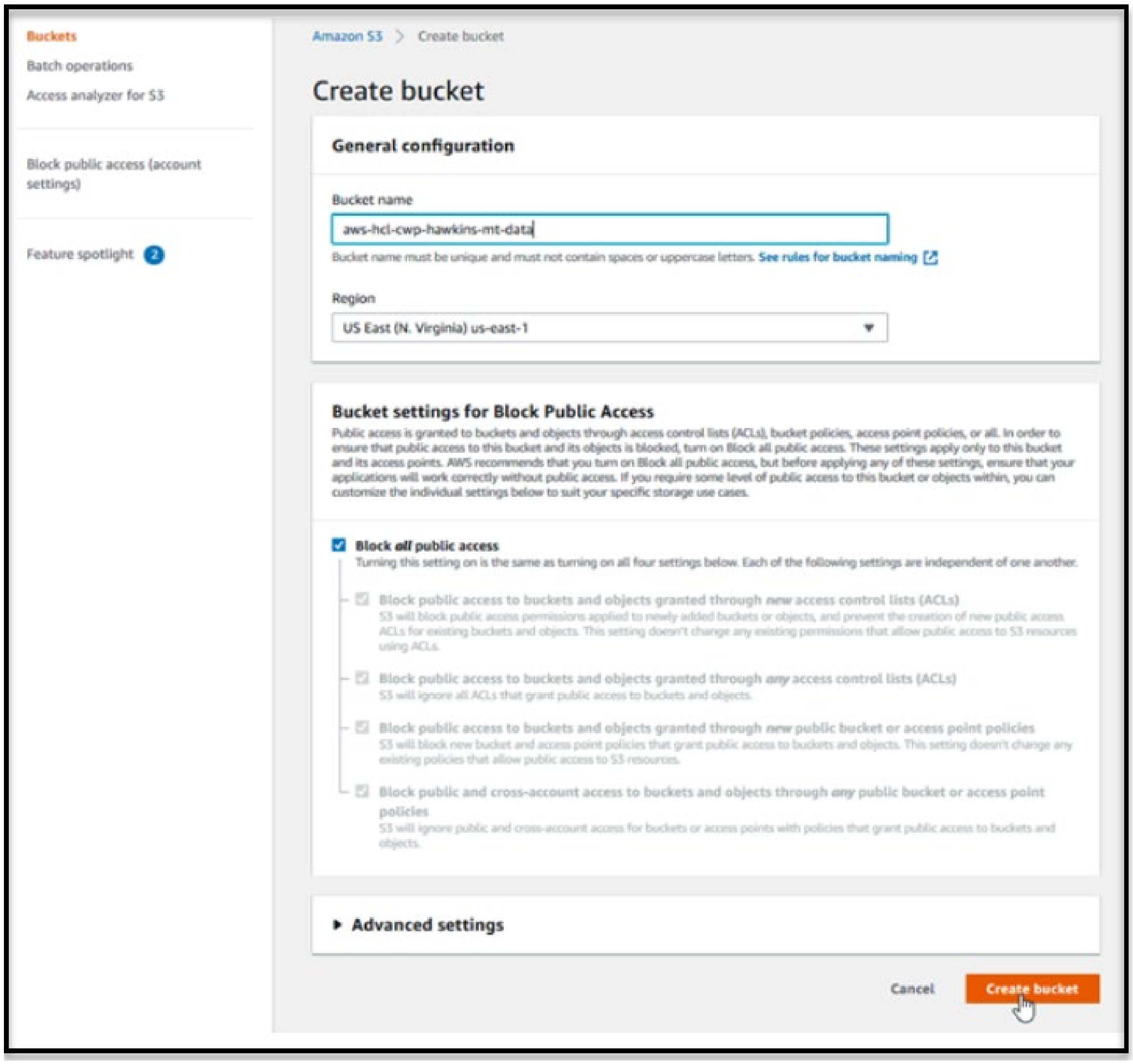
Prerequisite: An AWS account.

1. Log in to the AWS console and open the Buckets page:

   https://s3.console.aws.amazon.com/s3/home

2. Click **Create bucket**.

3. Specify a bucket name and make sure **Block all public access** ischecked.



Create bucket

General configuration

Bucket name: aws-hcl-cwp-hawkins-mt-data

Bucket name must be unique and must not contain spaces or uppercase letters. See rules for bucket naming

Region: US East (N. Virginia) us-east-1

Bucket settings for Block Public Access

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases.

✓ **Block *all* public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ✓ Block public access to buckets and objects granted through *new* access control lists (ACLs)
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ✓ Block public access to buckets and objects granted through *any* access control lists (ACLs)
  S3 will ignore all ACLs that grant public access to buckets and objects.
- ✓ Block public access to buckets and objects granted through *new* public bucket or access point policies
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ✓ Block public and cross-account access to buckets and objects through *any* public bucket or access point policies
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

4. Use the AWS command line interface to create two new policies, for writers and readers.

   a. Paste the JSON below into two text files, one for each policy. In the Resource section, replace "aws-hcl-cwp-hawkins-mt-data" with your bucket name.

   data-writers policy:
   ```
   {
      "Version": "2012-10-17",
      "Statement": [
         {
   ```

```
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListBucketVersions"
            ],
            "Resource": [
                "arn:aws:s3:::aws-hcl-cwp-hawkins-mt-data"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:PutObjectVersionAcl",

                "s3:AbortMultipartUpload"
            ],
            "Resource": [
                "arn:aws:s3:::aws-hcl-cwp-hawkins-mt-data/*"
            ]
        }
    ]
}
```

data-readers policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::aws-hcl-cwp-hawkins-mt-data"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::aws-hcl-cwp-hawkins-mt-data/*"
            ]
        }
    ]
}
```

b. Save the files as writers.json and readers.json.

c. In a command prompt window, enter the following commands and ensure they run without error:

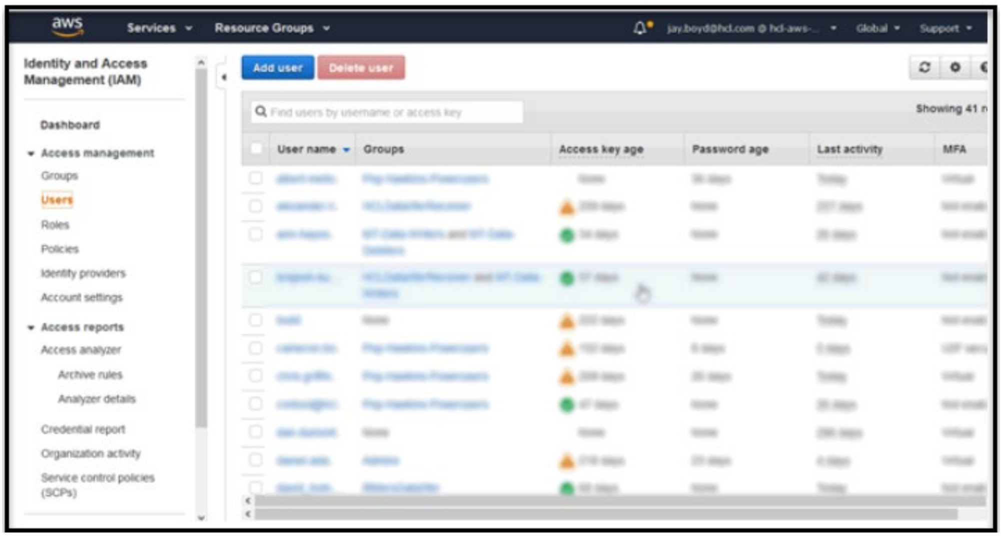
```
aws iam create-policy --policy-name my-policy-for-mt-data-writers --policy-document
file://writers.json
```

```
aws iam create-policy --policy-name my-policy-for-mt-data-readers--policy-document
file://readers.json
```

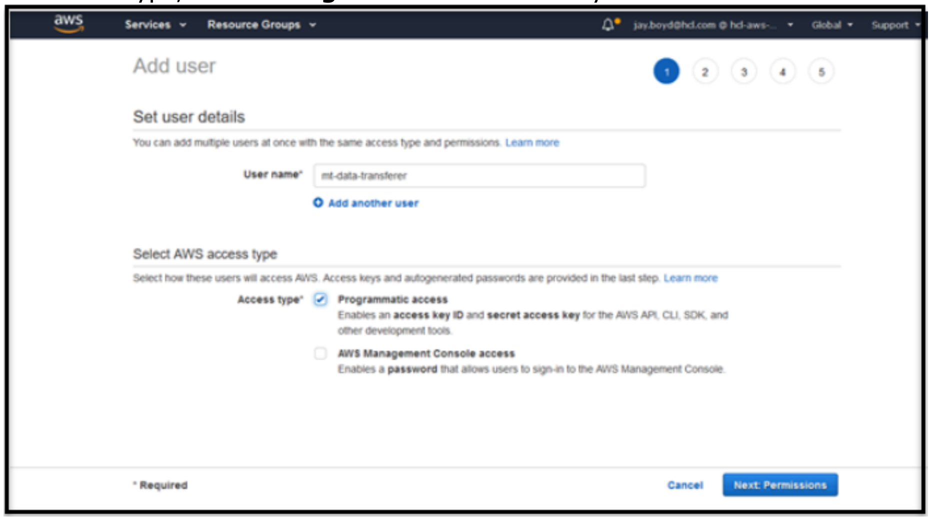
5. Go to the Identity and Access Management page:

   https://console.aws.amazon.com/iam/home

6. Click **Users** and then **Add user**.



7. Specify the user name and access type. User names may not contain spaces. They can contain alphanumeric characters, or any of the following: _+=,.@-. Be sure to add the user **ics-smartcloud-operations@wwpdl.vnet.ibm.com**. For the access type, choose **Programmatic access** only.



Add user

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* mt-data-transferer

Add another user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

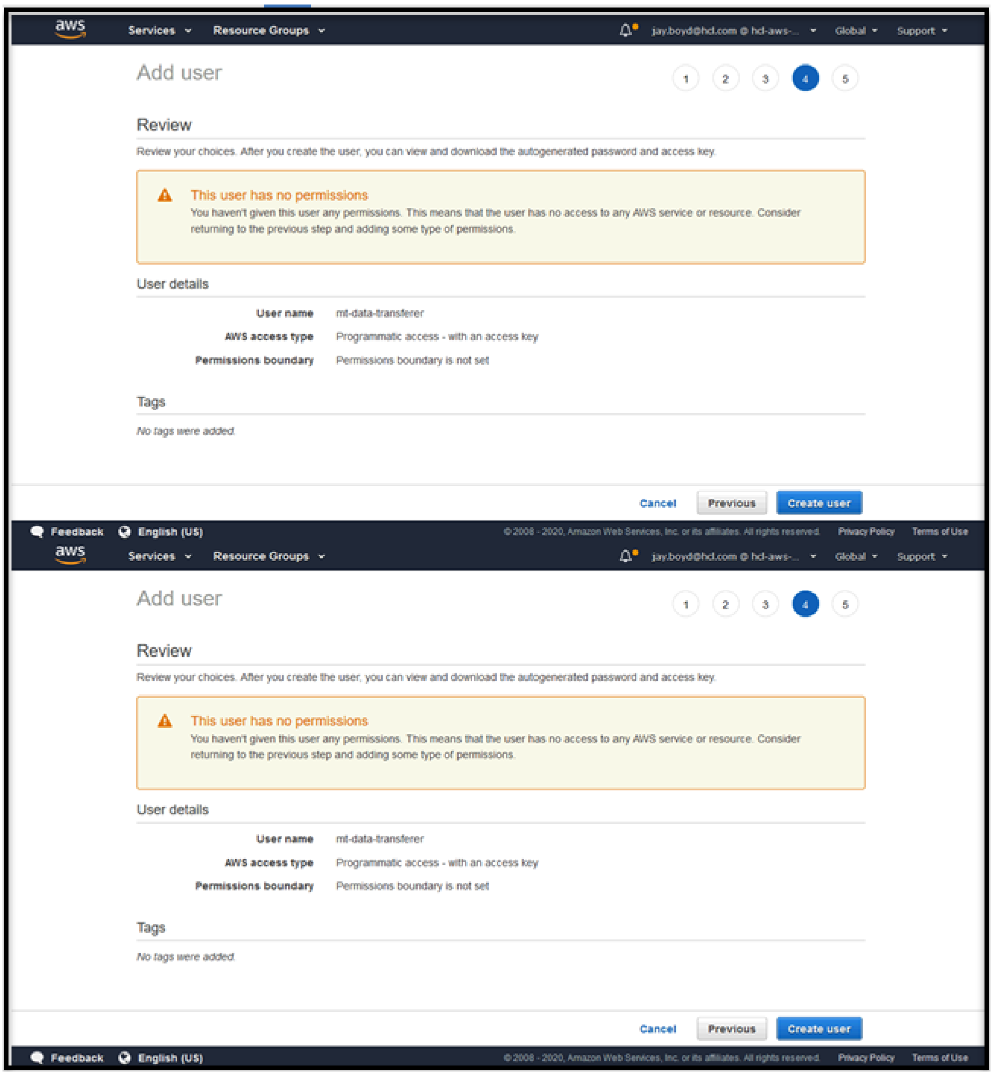
Access type* [✓] **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

[ ] **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

* Required

Cancel Next: Permissions

8. Click **Next** until you reach the Review screen. Don't add any privileges to the user. Privileges are defined in the policies you created. These are applied when you create user groups, attach the policies, and add users to the groups.



Add user

1 2 3 4 5

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

This user has no permissions
You haven't given this user any permissions. This means that the user has no access to any AWS service or resource. Consider returning to the previous step and adding some type of permissions.

User details

User name	mt-data-transferer
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Tags

No tags were added.

Cancel
Previous
Create user
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Add user

1 2 3 4 5

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

This user has no permissions
You haven't given this user any permissions. This means that the user has no access to any AWS service or resource. Consider returning to the previous step and adding some type of permissions.

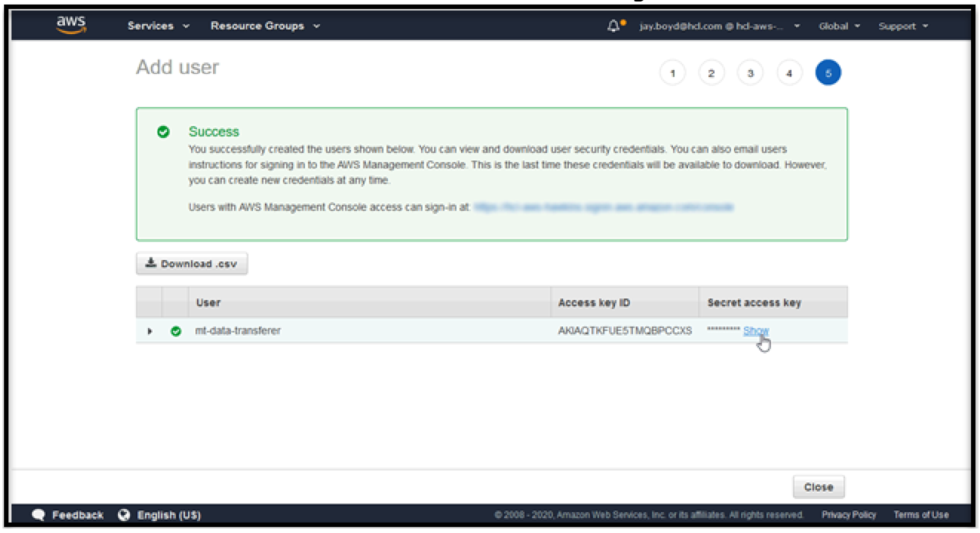
User details

User name mt-data-transferer
AWS access type Programmatic access - with an access key
Permissions boundary Permissions boundary is not set

Tags

No tags were added.

9. Click **Create user.** You should see a "success" message.



Add user

1 2 3 4 5

Success
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: [illegible]

User	Access key ID	Secret access key
mt-data-transferer	AKIAQTKFUE5TMQBPCCXS	******** Show

Close
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

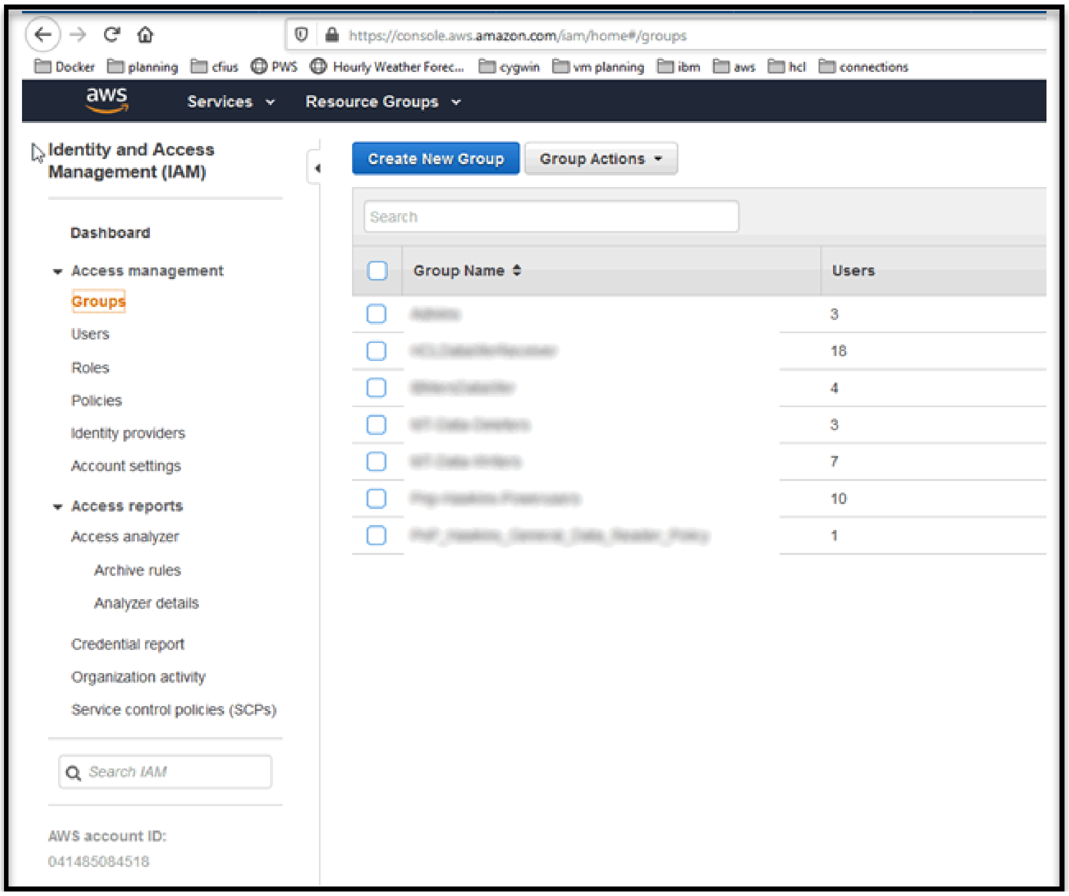
10. **Important:** Click **Show** and copy the access key id and secret access key displayed. You must give these to the user. You must also give the id and key for **ics-smartcloud-operations@wwpdl.vnet.ibm.com** to HCL Support. Once you leave this page you will not have access to the secret access key again.

11. Repeat as needed for other users.

12. For each policy that you created earlier, create a group and attach the appropriate policy to it. In the left-hand navigation, click

Groups.

13. Click **Create New Group**.



https://console.aws.amazon.com/iam/home#/groups

Docker | planning | cfius | PWS | Hourly Weather Forec... | cygwin | vm planning | ibm | aws | hcl | connections

Identity and Access Management (IAM)

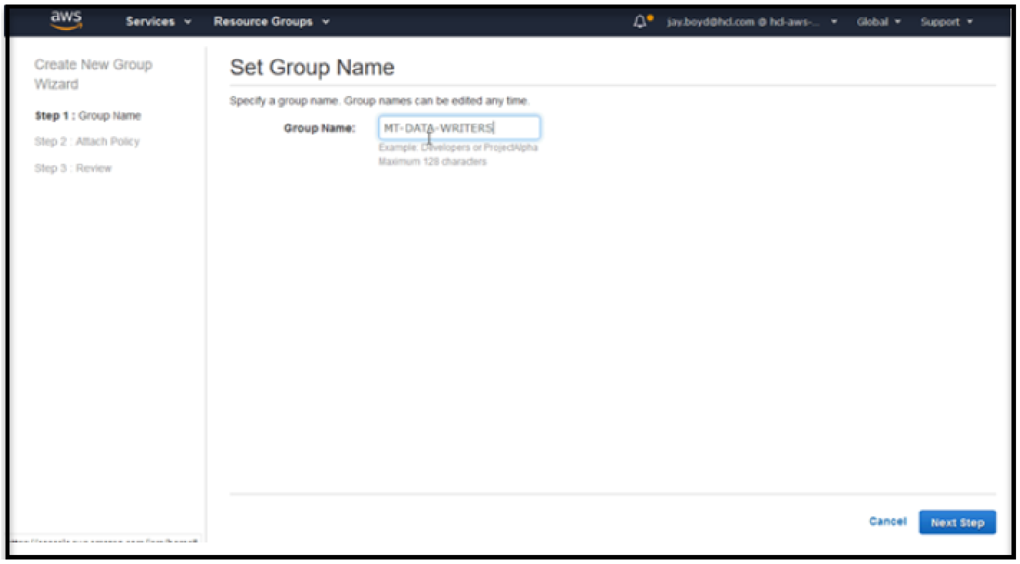
- Dashboard
- Access management
  - Groups
  - Users
  - Roles
  - Policies
  - Identity providers
  - Account settings
- Access reports
  - Access analyzer
    - Archive rules
    - Analyzer details
  - Credential report
  - Organization activity
  - Service control policies (SCPs)

Search IAM

AWS account ID:
041485084518

Group Name	Users
[redacted]	3
[redacted]	18
[redacted]	4
[redacted]	3
[redacted]	7
[redacted]	10
[redacted]	1

14. Use the group creation wizard to set the group name, attach the correct policy, and review your choices.

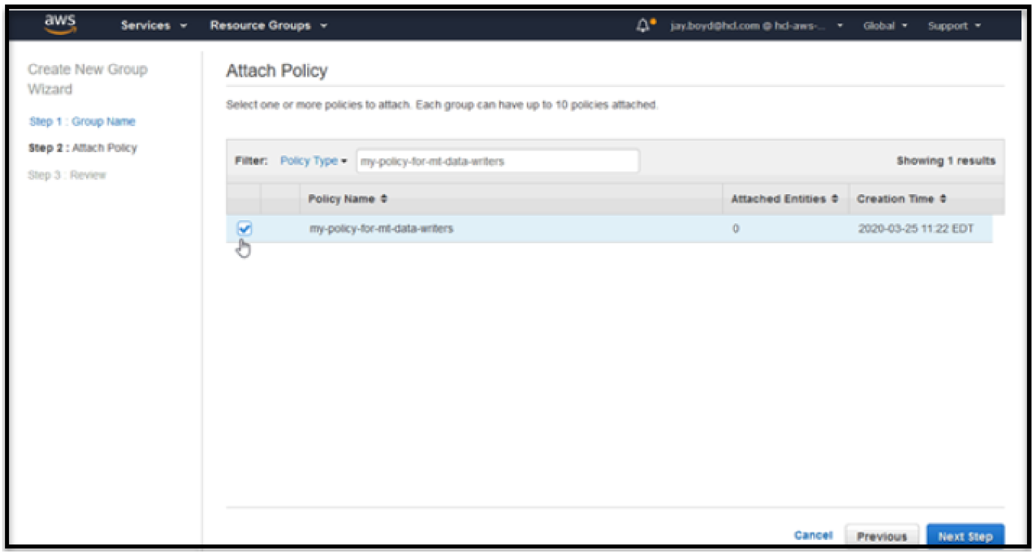


Set Group Name

Specify a group name. Group names can be edited any time.

Group Name: MT-DATA-WRITERS

Example: Developers or ProjectAlpha
Maximum 128 characters

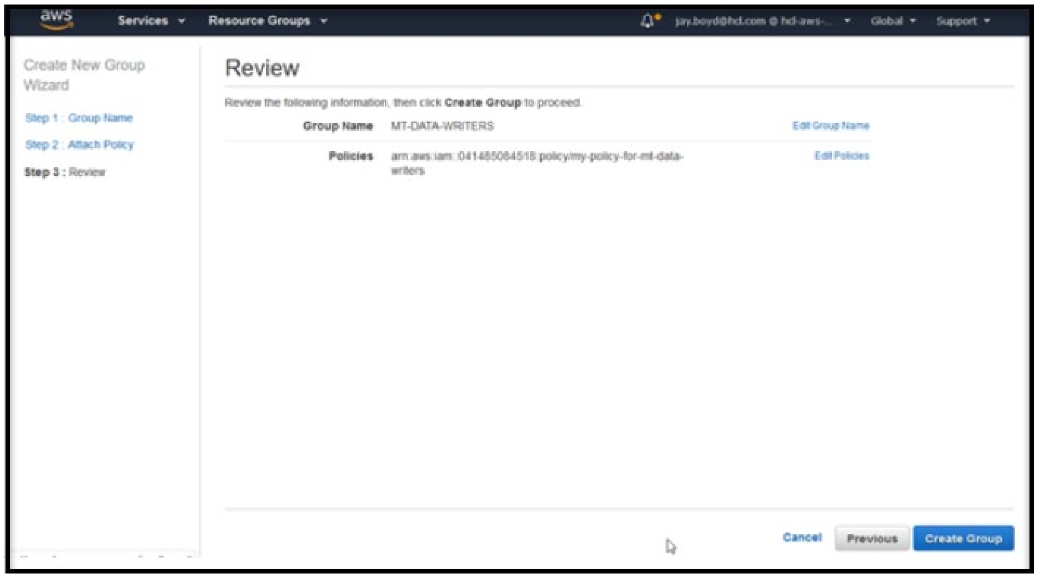


Attach Policy

Select one or more policies to attach. Each group can have up to 10 policies attached.

Filter: Policy Type ▾ my-policy-for-mt-data-writers — Showing 1 results

	Policy Name	Attached Entities	Creation Time
✓	my-policy-for-mt-data-writers	0	2020-03-25 11:22 EDT



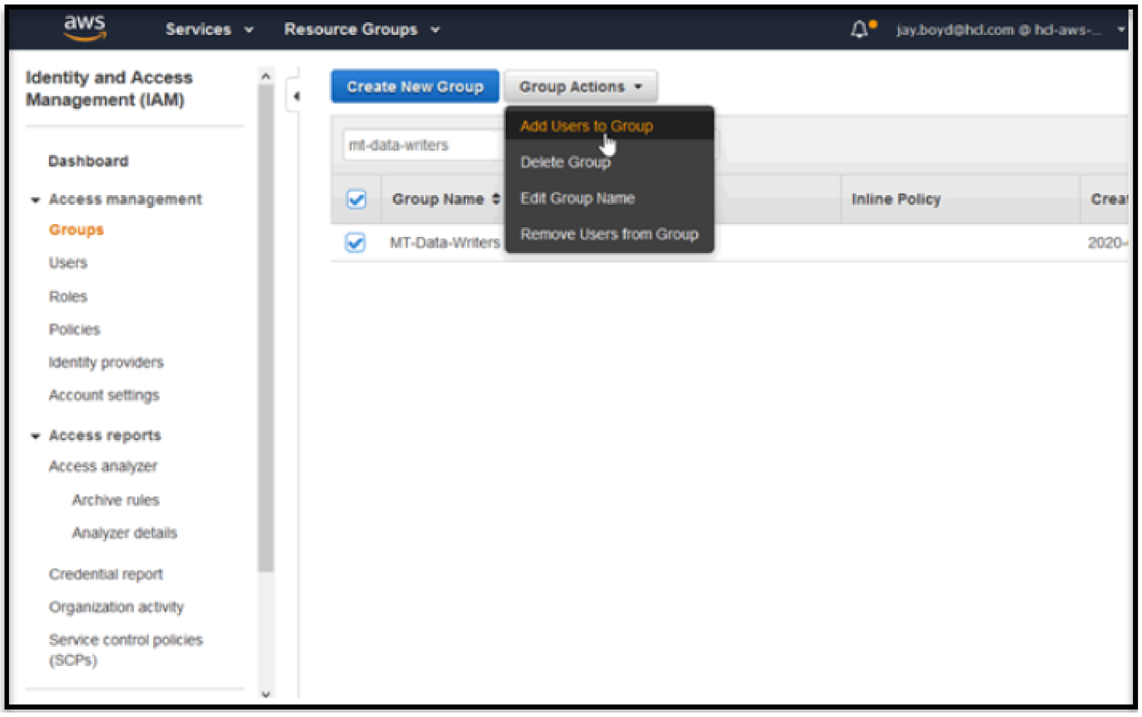
Review

Review the following information, then click **Create Group** to proceed.

Group Name	MT-DATA-WRITERS
Policies	arn:aws:iam::041485084518:policy/my-policy-for-mt-data-writers

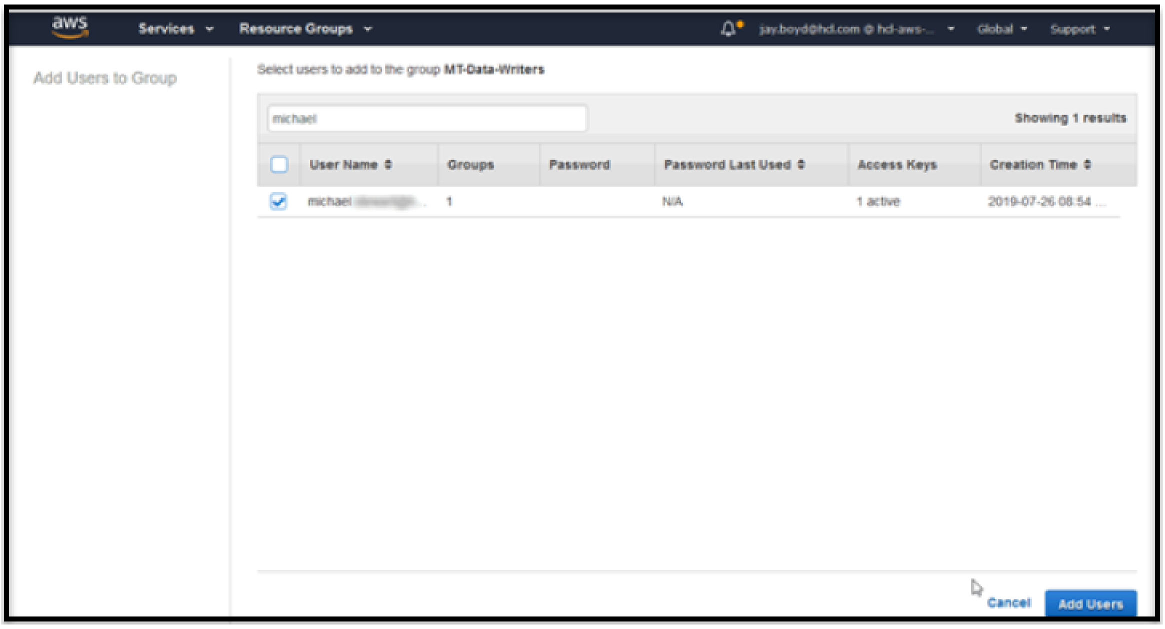
15. When you’re satisfied with your choices, click **Create Group.**
16. Repeat the group creation steps for the other policy you created.

17. Add users to your groups. Select a group and click **Group Actions > Add Users to Group**.



	Group Name	Inline Policy	Crea
✓	MT-Data-Writers		2020-

18. Select users and click **Add Users.** Add ics-smartcloud-operations@wwpdl.vnet.ibm.com to the data writers group and the data readers group.



Select users to add to the group **MT-Data-Writers**

Showing 1 results

	User Name	Groups	Password	Password Last Used	Access Keys	Creation Time
✓	michael [redacted]	1		N/A	1 active	2019-07-26 08:54 ...

19. Repeat the user addition for the other group.

TXT config file and Rclone parameters

The requirements that have been put in place for the AWS bucket and rclone encryption attributes are intended to support a transfer process that is primarily intended to protect the transfer of the customer's data between IBM and the customer or the customer's designated agent. The strict specification may seem rigid, but it is also, meant to facilitate the export and transfer process.

The bucket and encryption attributes:

1. The file must be named: transfer-config-<org_id>.txt. For example, transfer-config-1000530836.txt

2. The format of the file is (all fields must be

   included): region = <AWS data center

   region> access_key_id = <AWS S3 Access

   Key Id>

   secret_access_key = <AWS S3 Secret Access Key>

   bucket_name = <AWS Bucket Name>

   org_id = <Org ID>

   password = <***encrypted*** rclone encryption password>

   password2 = <***encrypted*** rclone encryption password

   2>

From this configuration specification, at the export side an S3 remote and a rclone crypt will be added to the rclone.conf file. The export process will specify the configured crypt to which all artifacts will be copied. The AWS bucket properties and keys are available to you as a consequence of having set up the bucket and created the policies and access key information.

Just as, the export side creates an S3 remote and a rclone crypt; the import side must create those as well. They can be created interactively by typing rclone config at the command line and following the steps (just keep in mind file names and directories are not be encrypted). Or you can issue the following commands at the command line or in a script to create the remote and the crypt. (For more details see https://rclone.org/commands/rclone_config_create/ and https://rclone.org/crypt/

1. Create the remote
   rclone config create <remote_name> **s3 provider** *AWS* **env_auth** *false* **access_key_id** <access_key_id> **secret_acces_key** <secret_access_key> **region** <region> **bucket_acl** *private* **disable_checksum** *false*

2. Create the crypt
   rclone config create <crypt_name> **crypt remote** <remote> **filename_encryption** *off* **directory_name_encryption** *false* **password** <your_password> **password2** <your_password2>

When you create the crypt using the command line, it will generate a stanza in your rclone.conf file for the crypt, with the encrypted passwords. It is those encrypted password strings that you must provide in the transfer config file, unencrypted passwords will not be accepted. Also, note while the crypt documentation states that password2 is optional, it is not optional for the transfer process.

How to set an org to Read-Only

When you are ready to request the final DB export and update binary files it is recommended that you set the org to read only. This can be performed by an admin of the Org using the customizer tool. Instructions for setting an org to read- only can be found here: https://github.com/ibmcnxdev/global-samples/tree/master/disable-content-creation

Downloading the exported data

Background

The Connections data for the hosted organizations (exported database and binary files) are available to be downloaded from the IBM cloud and can be imported into your Connections MT environment. Database and file exports will be available for download from an AWS S3 provided by the CH-MSP. Download and installation instructions are provided in this document. All artifacts are uploaded and are to be downloaded using the rclone tool. This document discusses the processes for downloading and importing the Connections data artifacts. A separate document has been provided with the details for setting up the Amazon s3 and how that information is to be conveyed to IBM.

Downloading the export "package"

Before you begin

Please see the document that describes how to set up an S3 bucket and convey the bucket information to IBM for data transfer.

About this task

As part of the transfer process, a directory will be created in the target bucket folder that identifies the organization for which exports are being transferred. The name of the directory is the orgid.

The files that are written to that folder in the bucket will be encrypted using an rclone "crypt." The keys used to encrypt the uploaded content are provided to IBM by the CH-MSP. The files must be downloaded using a corresponding crypt at the CH-MSP.

For each organization that is being migrated to the CH-MSP environment is a set of Db2 Export files, consisting of .ixf files as well as any binary large object files, or .lob files. The database export files will be found in directories named for the database that has been exported.

These directories are ACTIVITIES, BLOGS, CALENDAR, EMPINST, FILES, FORUM, HOMEPAGE, SNCOMM, WIKIS, and XCC. The .ixf and .lob files in each directory generally correspond to that database's tables. The database exports are found underneath a directory called db-exports on AWS S3 under an org specific directory.

In addition to those database export files, there is a set of binary files that represent the on-disk content managed by each application. Those applications are Files, Wikis, Activities, Forums, and Blogs.

For Forums, Activities, and Blogs, the binary files represent attachments to the content for those applications. For Wikis, the on-disk binary content represents Wiki pages as well as attachments to Wikis. For the Files application, the on-disk files represent all the versions of the files as well as accompanying thumbnails for those files.

File exports can be found in compressed archives under the organization specific file-exports directory, in subdirectories that correspond to each application. The use of tarballs to transfer the files was introduced to more efficiently transfer the files using rclone.

In each application-specific subdirectory there should be one archive file (e.g. wikis.tar.gz), a timestamp file, and if appropriate a file that contains a list of files for which there were references in the database but which could not be found on disk at export time. The exception is that in the Files application subdirectory there will be three archive files (e.g., files1.tar.gz, files2.tar.gz and files3.tar.gz), which correspond to the storage structure in the Cloud production environments.

Once the export package has been downloaded to a location in the MSP environment, the imports can be performed.

The Files exports should be copied from the S3 bucket to a staging directory, expanded, and then copied into the locations where Connections has been configured to store them.

Each of the applications stores files in different locations. The directories are configured as WebSphere environment variables in the WebSphere admin console.

Importing the binary files

Directory layouts for files and attachments

All applications, with the exception of blogs and some activities, use the same scheme for naming on-disk files. File names are globally unique using a UUID format where a directory structure of a depth of two is calculated from the UUID name. You will note that the two levels of directory are named from 0-128. This scheme is used to evenly distribute files across the directory tree. Files, Wikis, and Forums use this approach for all files and the GUID references are present in the DB tables, so their on-disk location is easily calculated.

Activities uses the same approach that Files, Wikis, and Blogs uses for "newer" files under a directory called "activityobjectstore4." You might or might not note that there are directories activityobjectstore1-3 under which there are directories and files that use a different naming scheme. That scheme was deprecated at some point in the past, though you might see some files that were created when that scheme was in use. Activities has references in its database tables for all of the file attachments.

For Blogs, each "website" object, which is effectively the blog container object, maintains a correspondingly named GUID directory that can be found under directory structure of a depth of three that are named for the first three characters in that GUID name. For example, 4C2617BB-12D2-404D-9D7F-D0EC1D76ADE5 would be found under a directory of "4/C/2" (case insensitive). Blogs can also have attached folders in addition to attached files, and those directory names, which are not UUID based, use that same 3 depth directory naming scheme: a folder named "MyPhotos" would be found under "m/y/p". Because the naming scheme is somewhat less predictable, we have added each directory from the top level as a tarball itself to package things up neatly. It appears that the Blog application might have created directories that have no subfolders or files in them. For the sake of referential integrity, we tar those up as well. Aside from the Website UUID, there are no additional file references in the Blogs tables. You also might find a blogs_nf.txt file; those are very likely false positives in which there might have been files that were not found on the disk volumes where cloud files had been stored.

Below are examples for copying each of the files for each application to the appropriate location. Note that organizational separation is maintained in the database tables, but all files are mixed when on the file system.

Activities

1. ACTIVITIES_CONTENT_DIRECTORY: /opt/HCL/data/shared/activities/content
2. Expand Activities archive to the Activities staging directory.
3. Copy the files to the ACTIVITIES_CONTENT_DIRECTORY, as follows:

   ```
   cp -vrT <staging directory>/file-exports/activities/activitiesobjectstore4 /opt/HCL/data/shared/activities/content
   ```

   ```
   cp -vrT <staging directory>/files-exports/activities/activitiesobjectstore1 /opt/HCL/data/shared/activities
   ```

Blogs

1. BLOGS_CONTENT_DIRECTORY: /opt/HCL/data/shared/blogs/upload/content
2. Expand Blogs archive to the Blogs staging directory.
3. Because Blogs directories are also archived, perform the following command to expand the Blogs directories and clean them up:

   ```
   find . -type f -name '*.tar.gz' -exec sh -c 'dir="$(dirname ""{}"")"; tar xvf "{}" -C "${dir}"; rm "{}" ' \;
   ```

4. Copy the files to the BLOGS_CONTENT_DIRECTORY as follows:
   cp -vrT <staging directory>/file-exports/blogs /opt/HCL/data/shared/blogs/upload/content

Files

1. FILES_CONTENT_DIRECTORY: /opt/HCL/data/shared/files/upload/files
2. Expand files archive files to the files staging directory:

   cp -vrT <staging directory>/file-exports/files /opt/HCL/data/shared/files/upload/files

Forums

- FORUMS_CONTENT_DIRECTORY: /opt/HCL/data/shared/forums/content
- Expand forums archive file to the forums staging directory:

  cp -vrT <staging directory>/file-exports/forums /opt/HCL/data/shared/forums/content

Wikis

1. WIKIS_CONTENT_DIRECTORY: /opt/HCL/data/shared/wikis/upload/files
2. Expand wikis archive file to the wikis staging directory:

   cp -vrT <staging directory>/file-exports/wikis /opt/HCL/data/shared/wikis/upload/files