Clowder

Release 1.21.0

Luigi Marini

Sep 08, 2022

GENERAL

Welcome to Clowder's documentation. Clowder is a web-based data management system that allows users to share, annotate, organize and analyze large collections of datasets. It provides support for extensible metadata annotation and distributed analytics for automatic curation of uploaded data. Clowder is open source software that can be customized and deployed on your own cloud.

> **Warning:** This documentation is being updated. Please bear with us as we make this a much more useful document. If you want to contribute to the documentation the source is available here. Thank you!

Note:

- For a high level overview of the project please visit https://clowderframework.org.

- If you have questions for the Clowder you can chat with the developers on Slack or by sending an email to the mailing list *clowder@lists.illinois.edu* (sign up).

- Issue tracking, internal documents, continuous build and other information is available on NCSA Opensource. A copy of the source code can also be found on GitHub.

- We are always looking for contributions.


CONTENTS

1.1 Overview

Clowder has been build from the ground up to be easily customizable for different research and applications domains. Here some of the reasons why you would want to adopt it:

- You want both an extensive web interface and web service API to be able to easily browse the data in the web browser and also script how you manipulate the data.

- You want to customize how you preview, store and curate the data.

- You have code you want to run on a subset of the data as it is ingested into the system.

- You have a lot of data and prefer hosting the data yourself instead of paying for cloud storage solutions.

There is no single Clowder instance. Clowder is open source software that can be installed and maintained by individual users, research labs, data centers.

1.1.1 Data Model

The basic data model is very generic to support many different cases. This means that specific communities will have to adopt and customize how the different resource types are used within a specific use case.
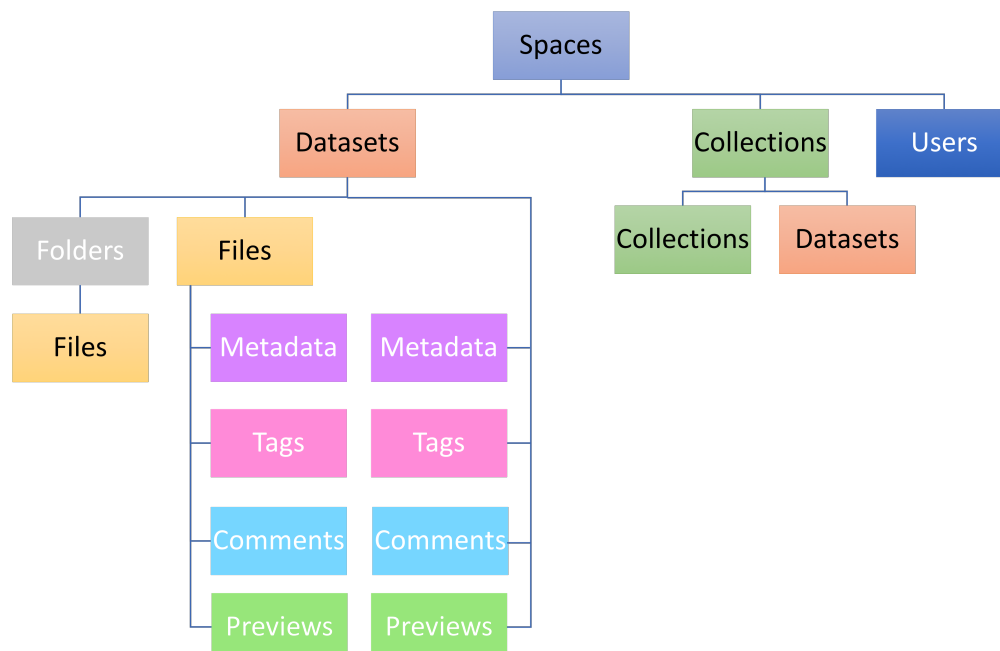
1.2 Changelog

1.2.1 Change Log

All notable changes to this project will be documented in this file.

The format is based on Keep a Changelog and this project adheres to Semantic Versioning.

1.21.0 - 2022-08-23

Important: This update requires a MongoDB update schema due to a bug in the original migration of showing summary statistics at the space level. Make sure to start the application with -DMONGOUPDATE=1. You can also run the fixCounts.js script prior to upgrading to minimize the downtime.

Added

- api.Files jsonfile, adds two fields "downloads" and "views" #228

- Dataset and file scala.html pages incl schema.org jsonld metadata for (google)datasetsearch #335

- MiniUser and LicenseData now have to_jsonld methods to return string part of #335 metadata

- LicenseData has urlViaAttributes used by it's to_jsonld to guess url when empty, for #335

- MRI previewer for NIFTY (.nii) files.

- Dataset page usually defaults to Files tab, but if no files will now show Metadata first

- HEIC (.heic) and HEIF (.heif) mimetypes to support new Apple iPhone image file format.

- In the docker container the folder /home/clowder/data is now whitelisted by default for uploading by reference. This can be changed using the environment variable CLOWDER_SOURCEPATH.

- The current CLA for developers of clowder.

Fixed

- Send email to all admins in a single email when a user submits 'Request access' for a space

- Send email to all admins and request user in a single email when any admin accepts/rejects 'Request access' for a space #330

- script/code to count space in files was not correct #366

- github actions would fail for docker builds due to secrets not existing

- Fix to remove dataset from a space #349

Changed

- Utils.baseURL now on RequestHeader instead of Request[Any]
- MongoDB Service log error:'Not all dataset IDs found for Dataset|Folder bulk get request', now incl all the IDs notFound

1.20.3 - 2022-06-10

Fixed

- docker images are build using newest java version #361

Changed

- docker builds images for amd64 and arm64 #322

1.20.2 - 2022-04-30

Fixed

- swagger lint action
- When downloading a file with a `'` in the name it would save the file as blob
- Fix for a rare race condition with masonry where tiles could end up overlapping in space page.
- Fixes bug where same extractor shows up multiple times and all Clowder instances index db on reindex #327

Changed

- Changed `Enabled By SuperAdmin` to read `Enabled by Server Admin` #344

1.20.1 - 2022-04-04

Fixed

- Fix to incorrect sorting for default cases. #248
- Masonry package was not loaded from right location sometimes.
- Fixes bug where "Select a Collection" Parent Collection dropdown on Collection page was not working due to missing parameter #343

Added

- Documentation: Installing Clowder on Apple Silicon M1
- Documentation: Customizing clowder's deployment, simplified duplicate instructinos.
- Documentation: Added "How to contribute documentation" page
- Documentation: New Sphinx plugins for dropdowns and menus.

1.20.0 - 2022-02-07

Added

- An IFC previewer

Fixed

- Conf file and code had incosistent spelling of BagIt. Now all have capital B and I.
- When event stream is disabled don't show for logged in user #280
- three.js is no longer associated with application/octet-stream, now with models #305

Changed

- Download of dataset/collection now has optional parameter bagit (default false) to download items in bagit format.
- The FBX previewer can also load GLTF files

1.19.5 - 2022-01-21

Fixed

- Removed JMSAppender and SocketServer from log4j.
- Cleaned up getting started documentation

1.19.4 - 2021-11-11

Fixed

- Error with adding dataset to collection. #306

1.19.3 - 2021-11-11

See fix-counts.js for a script that can be run before this update (as well as the update for 1.19.0) to pre populate the migration. This will speed up the update and will not impact the running instance.

Fixed

- If a space has a lot of datasets, rendering the space page is very slow. Files in a space is now cached.
- Set permissions for folders to be 777 this fixes clowder-helm#5

1.19.2 - 2021-10-20

Fixed

- Error with library dependencies broke search capabilities, rolled back to known working versions

1.19.1 - 2021-10-19

Added

- Support the DefaultAWSCredentialsProviderChain for passing in credentials to the S3ByteStorageService.

Fixed

- Cleaning up after a failed upload should no longer decrement the file + byte counts.
- Fix the broken preview after file deletion within a folder. #277
- Fix public spaces not displaying correctly if not logged in.

Changed

- Now building mongo-init and monitor docker containers with python 3.8
- Upgraded extractor parameters jsonform to version `2.2.5`.

Removed

- Check image is now part of ncsa/checks

1.19.0 - 2021-10-05

Important: This update requires a MongoDB update schema due to the new ability of showing summary statistics at the space level. Make sure to start the application with -DMONGOUPDATE=1.

Fixed

- Adding dataset to space. Space list on dataset page would be empty - fixed error when no spaces would load. #274
- Typos "success" when returning status from API and "occurred" when logging to console.
- If a dataset had multiple folders the layout would be wrong.
- Collections created using api route are now indexed upon creation. #257

Added

- Mime type for geojson
- "when" parameter in a few GET API endpoints to enable pagination #266
- Show space statistics (bytes, users. etc) #119
- "id" in GET metadata.jsonld endpoints #278
- 'POST /api/files/uploadToDataset' now allows folder_id for uploading file to folder. #232

1.18.1 - 2021-08-16

This release fixes a critical issue where invalid zip files could result in the files not being uploaded correctly. To check to see if you are affected, please use the following query:

```
db.uploads.find({"status": "CREATED", "contentType": "application/x-zip-compressed"},
→{"author.fullName": 1, "author.email": 1, "filename": 1, "uploadDate": 1, "length":␣
→1})
```

If any files are returned, you should check to see if these files affected and are missing from clowder.

Fixed

- When zip file is uploaded, it will parse the file to check if it is a valid zip file, this couuld result in files not stored in final storage space #264
- Updated swagger documentation
- Return 404 not found when calling file/dataset/space api endpoints with an invalid ID #251
- Line breaks in welcome message breaks swagger build #187

Changed

- Added more information when writing files to make sure files are written correctly
- Made cilogon group check debug message instead of error message

1.18.0 - 2021-07-08

Added

- Added folder and folder id to API call `GET /api/datasets/:id/files`. #34
- Ability to queue archive / unarchive for full datasets.
- API status endpoint `GET /api/status` will now show what storage type is used and for superadmins will show more information about the backend storage.
- `GET /api/files/bulkRemove` now returns status of files deleted, not found, no permission, or errors.

Fixed

- When uploading a file, any extractors marked disabled at the space level would be ignored. #246
- RabbitMQ will not use connection if it does not exist.
- Previews returns 404 if preview is not found `GET /api/previews/:id`.
- Added index for comments, will speed up index creation.
- If using S3 storage in docker, it was not reflected correctly in the docker-compose file.
- Docker image for mongo-init now based on python:3.7-slim to reduce size.

1.17.0 - 2021-04-29

Fixed

- Close channel after submitting events to RabbitMQMessageService.

Added

- Endpoint `/api/datasets/createfrombag` to ingest datasets in BagIt format. Includes basic dataset metadata, files, folders and technical metadata. Downloading datasets now includes extra Datacite and Clowder metadata.
- Endpoint `/api/files/bulkRemove` to delete multiple files in one call. #12
- Log an event each time that a user archives or unarchives a file.
- Persist name of message bus response queue, preventing status messages from getting lost after a reboot.

Changed

- Updated Sphinx dependencies due to security and changes in required packages.
- Updated the three.js libraries for the FBX previewer

1.16.0 - 2021-03-31

Fixed

- Remove the RabbitMQ plugin from the docker version of clowder

Added

- Added a `sort` and `order` parameter to `/api/search` endpoint that supports date and numeric field sorting. If only order is specified, created date is used. String fields are not currently supported.
- Added a new `/api/deleteindex` admin endpoint that will queue an action to delete an Elasticsearch index (usually prior to a reindex).
- JMeter testing suite.

Changed

- Consolidated field names sent by the EventSinkService to maximize reuse.
- Add status column to files report to indicate if files are ARCHIVED, etc.
- Reworked auto-archival configuration options to make their meanings more clear.

1.15.1 - 2021-03-12

Fixed

- Several views were throwing errors trying to access a None value in `EventSinkService` when a user was not logged in. Replaced `get()` with `getOrElse()`.
- Consolidated field names sent by the EventSinkService to maximize reuse.
- Changed `EventSinkService` logging to debug to minimize chatter.
- Don't automatically create eventsink queue and bind it to eventsink exchange. Let clients do that so that we don't have a queue for the eventsink filling up if there are no consumers.

1.15.0 - 2021-03-03

Added

- CSV/JSON previewer using Vega.

- Previewer for FBX files.

- `created` search option for filtering by upload/creation date of resource.

- `EventSinkService` to track user activity. All events are published to the message queue. Multiple consumers are available in event-sink-consumers.

Fixed

- Clowder will no longer offer a Download button for a file until it has been PROCESSED.

- When space created through api the creator was not added to space as admin #179.

Changed

- `/api/me` will now return some of the same information as response headers. Can be used by other services to single sign on when running on same host.

- `RabbitMQPlugin` has been split into `ExtractorRoutingService` and `MessageService` to isolate the rabbitmq code from the extraction code.

Removed

- the toolserver is no longer build as part of clowder since it is no longer maintained. We are working on a newer version that will be included in future versions of clowder.

1.14.1 - 2021-02-02

- Google will no longer work as login provider, we are working on this issue #157.

- If non local accounts are used the count can be wrong. Use the fixcounts script to fix this.

Fixed

- Error logging in with Orcid due to changed URL. #91

- Fixed error in url for Twitter login.

- Users count was not correct if using anything else but local accounts. #136

- Files were not properly reindexed when the Move button was used to move a file into or out of a folder in a dataset.

- When adding a file to a dataset by URL, prioritize the URL `content-type` header over the file content type established by looking at the file name extension. #139

- Wrap words across lines to stay within interface elements. #160

1.14.0 - 2021-01-07

Added

- Added a previewer for FBX files.

- Added a new `/api/reports/metrics/extractors` report for summarizing extractor usage by user. Database administrators can use `scripts/updates/UpdateUserId.js` to assign user IDs to older extraction event records based on resource ownership in order to improve the accuracy of the report for older data.

Changed

- `api/reports/storage/spaces` endpoint now accepts a space parameter for ID rather than requiring a space filter.

- Datasets and collections in the trash are no longer indexed for discovery in search services.

- Switched to loading the 3DHOP libraries used by `viewer_hop.js` from http://vcg.isti.cnr.it/3dhop/distribution to https://3dhop.net/distribution. The new server is a safer https server.

1.13.0 - 2020-12-02

Added

- Ability to submit multiple selected files within a dataset to an extractor.

- Support for Amplitude clickstream tracking. See Admin -> Customize to configure Amplitude apikey.

- UpdateUserId.js to scripts/updates. This code adds user_id to each document in extractions collection in mongodb. user_id is taken from author id in uploads.files if exists, else it taken from author id in datasets collection.

Fixed

- An extractor with file matching set to `*/*` (all file types) would incorrectly send out dataset events.

- Space Editors can now delete tags on files, datasets and sections.

- GeospatialViewer previewer no longer shows if file does not contain geospatial layers.

1.12.2 - 2020-11-19

Changed

- /api/reindex admin endpoint no longer deletes and swaps a temporary index, but reindexes in-place.

1.12.1 - 2020-11-05

Fixed

- Error uploading to spaces that did not have extractors enabled/disabled (personel spaces).

- If extractor does not have any parameters, there would be an error message in the console of the browser.

- If the extractor did not have a user_id it would create an error and not record the event.

Changed

- Docker Images are now pushed to github container registry

1.12.0 - 2020-10-19

Warning:

- This update modifies the MongoDB schema. Make sure to start the application with `-DMONGOUPDATE=1`.

- This update modifies information stored in Elasticsearch used for text based searching. Make sure to initiate a reindex of Elasticsearch from the Admin menu or by `POST /api/reindex`.

Added

- Global extractors page now shows more information, including submission metrics, logs (using Graylog), job history and extractors maturity. Extractors can be grouped using labels. User can filter list of extractors by labels, space, trigger and metadata key.

- Users have more refined options to set extractors triggers at the space level. They can now follow global settings, disable and enable triggers.

- Ability to set chunksize when downloading files. Set defult to 1MB from 8KB. This will result in faster downloads and less CPU usage at the cost of slightly more memory use.

- Support for parsing of Date and Numeric data in new metadata fields. New search operators <, >, <=, >= have been added to search API now that they can be compared properly.

- Track user_id with every extraction event. #94

- Added a new storage report at `GET api/reports/storage/spaces/:id` for auditing user storage usage on a space basis.

- The file and dataset metrics reports also have support for since and until ISO8601 date parameters.

- Added `viewer_hop` a 3D models previewer for `*.ply` and `*.nxz` files. Added `mimetype.nxz=model/nxz` and `mimetype.NXZ=model/nxz` as new mimetypes in `conf/mimetypes.conf`

Fixed

- Ignore the `update` field when posting to `/api/extractors`. #89
- Search results were hardcoded to be in batches of 2.
- Fixed permissions checks on search results for search interfaces that would cause misleading counts. #60

1.11.2 - 2020-10-13

Fixed

- Clowder healthcheck was not correct, resulting in docker-compose never thinking it was healthy. This could also result in traefik not setting up the routes.

1.11.1 - 2020-09-29

Added

- Added healtz endpoint that is cheap and quick to return, useful for kubernetes live/ready checks.

Fixed

- Fixed health check script when using custom path prefix.
- Proxy will no correctly handle paths that end with a / at the end.
- Submitting an extraction will always return a 500 error, see #84
- Added MongoDB index for `folders.files`.

Changed

- Updated update-clowder script to work with migration to github. Has the ability now to push a message to MSTEAMS as well as influxdb.

1.11.0 - 2020-08-31

Added

- Downloaded datasets now include DataCite v4 XML files in the output /metadata folder.
- Script to clean extractors' tmp files `scripts/clean-extractors-tmpfiles/`.
- Script for RabbitMQ error queue cleanup `scripts/rmq-error-shovel/`.
- Ability to use HTML formatting in the welcome message on the home page. #51
- Expose a read-only list of extractors to all users.

Changed

- Improved test script `scripts/tester/tester.sh` to report successes once a day.

Fixed

- Escape colon characters on search values for search box and advanced search to allow those values in a search.
- Typesafe now only offers https access. #49
- If uploading files by url > 2147483647 it would fail. #54

1.10.1 - 2020-07-16

Fixed

- Queue threads (e.g. Elasticsearch indexer) will no longer crash permanently if the queue connection to Mongo is lost temporarily.
- Docker images would not build correctly on GitHub.
- If monitor HTTP server would crash, it would not restart correctly.
- Don't call server side twice when rendering list of files on dataset page. #7
- Fixed Sphinx build errors and switched to using pipenv. Now building docs on readthedocs.

Added

- GitHub artifacts can be uploaded using SCP to remote server.

1.10.0 - 2020-06-30

Added

- Ability to mark multiple files in a dataset and perform bulk operations (download, tag, delete) on them at once.

Fixed

- Return thumbnail as part of the file information. #8
- Datasets layout on space page would sometimes have overlapping tiles.

Changed

- mongo-init script with users would return with exit code -1 if user exists, now returns exit code 0.

1.9.0 - 2020-06-01

Warning: This update modifies information stored in Elasticsearch used for text based searching. To take advantage of these changes a reindex of Elasticsearch is required. A reindex can be started by an admin from the Admin menu.

Added

- Ability to delete extractor, both from API and GUI. CATS-1044

- Add tags endpoint now returns the added tags. CATS-1053

- Ability to search by creator name and email address for all resources.

- List Spaces/Datasets/Collections created by each user on their User Profile page. CATS-1056

- Allow user to easily flip through the files in a dataset. CATS-1058

- Ability to filter files and folders in a dataset when sorting is enabled.

- Visualize existing relations between datasets on the dataset page. This can be extended other resource types. CATS-1000

- S3ByteStorageService verifies bucket existence on startup and creates it if it does not exist. CATS-1057

- Can now switch storage provider in Docker compose, for example S3 storage. See env.example for configuration options.

- Script to test extractions through the API.

Fixed

- When adding tags to a section of an image, show the new tag without having to refresh the page. CATS-1053

Changed

- Removed buttons to remove datasets from spaces and collections from certain pages. Moved Remove button for subcollections to right side of page to be consistent with other pages. CATS-1055

- Upgraded swagger documentation to openapi v3.0.1.

1.8.4 - 2020-05-15

Warning: This update modifies how information is stored in Elasticsearch for text based searching. To take advantage of these changes a reindex of Elasticsearch is required. This can be started by an admin either from GUI or through the API.

Fixed

- Fixed a bug related to improper indexing of files in nested subfolders, which could also affect searching by parent dataset.

1.8.3 - 2020-04-28

Warning: This update modifies how information is stored in Elasticsearch for text based searching. To take advantage of these changes a reindex of Elasticsearch is required. This can be started by an admin either from GUI or through the API.

Changed

- Elasticsearch indexer will now store new metadata fields as strings to avoid unexpected behavior on date fields.
- When reindexing use a temporary index to reindex while the current one is in use then swap.

Fixed

- Ability to delete tags from sections and files on the file page. CATS-1046 CATS-1042
- User-owned resources will now appear in search results regardless of space permissions.
- Updating space ownership for datasets and collections will correctly reindex those resources for searches.
- Missing index in statistics which would slow down system when inserting download/views.

Added

- GitHub Actions to compile and test the code base, create documentation and docker images.
- Code of Conduct as MD file (will be displayed by GitHub).
- Templates for Bug, Feature and Pull Request on GitHub.

1.8.2 - 2020-02-19

Fixed

- Use the passed-in length within S3ByteStorageService.save.

1.8.1 - 2020-02-05

Removed

- Removed unused RDF libraries. This was probably used by the rdf/xml functionality that was removed a while back but the dependencies were never removed.

- Removed Jena validation of JSON-LD metadata. It was creating a blank graph and clients couldn't upload metadata when Clowder runs in a location that doesn't not have access to https://clowderframework.org/contexts/metadata.jsonld.

Added

- Scripts to migrate files on disk AWS S3. CATS-1034

Changed

- Include collection prefix in path when saving to S3.

- Include length of file in `FileService` when saving the bytes to any backend service. This helps optimize S3 implementation.

- Upgraded sbt from 0.13.0 to 0.13.6 to fix build failures. CATS-1038

Fixed

- Calling api/Files.removeFile should no longer decrement related counters twice. CATS-929

1.8.0 - 2019-11-06

Warning: This update adds a new permission for archiving files and adds it to the Admin role. Please make sure to run Clowder with the ``MONGOUPDATE`` flag set to update the database.

Changed

- `/api/search` endpoint now returns JSON objects describing each result rather than just the ID. This endpoint has three new parameters - from, size, and page. The result JSON objects will also return pagination data such as next and previous page if Elasticsearch plugin is enabled and these parameters are used.

- S3ByteStorageService now uses AWS TransferManager for saving bytes - uploads larger than ~1GB should now save more reliably.

- `/api/search` endpoint now returns JSON objects describing each result rather than just the ID.

- Clean up docker build. Use new buildkit to speed up builds. Store version/branch/git as environment variables in docker image so that they can be inspected at runtime with Docker.

- Extractors are now in their own docker-compose file. Use Traefik for proxy. Use env file for setting options.

- Utilize bulk get methods for resources widely across the application, including checking permissions for many resources at once. Several instances where checks for resource existence were being done multiple times (e.g. in a method and then in another method the first one calls) to reduce MongoDB query load. These bulk requests will also report any missing IDs in the requested list so developers can handle those appropriately if needed.

- Removed mini icons for resource types on tiles. CATS-1031

- Cleaned up pages to list and enable extractors. Added description of what the page does. Added links to extraction info pages. Removed authors from table.

Added

- Ability to pass runtime parameters to an extractor, with a UI form dynamically generated UI from extractor_info.json. CATS-1019

- Infinite scrolling on search return pages.

- Trigger archival process automatically based on when a file was last viewed/downloaded and the size of the file.

- Script to check if mongodb/rabbitmq is up and running, used by Kubernetes Helm chart.

- Queuing system that allows services such as Elasticsearch and RabbitMQ to store requested actions in MongoDB for handling asynchronously, allowing API calls to return as soon as the action is queued rather than waiting for the action to complete.

- New extractors monitor docker image to monitor extraction queues. Monitor app includes UI that shows selected information about the extractors

- New `/api/thumbnails/:id` endpoint to download a thumbnail image from ID found in search results.

- New utility methods in services to retrieve multiple MongoDB resources in one query instead of iterating over a list.

- Support for MongoDB 3.6 and below. This required the removal of aggregators which can result in operations taking a little longer. This is needed to support Clowder as a Kubernetes Helm chart. CATS-806

- New Tree view as a tab in home page to navigate resources as a hiearchical tree (spaces, collections, datasets, folders and files). The tree is lazily loaded using a new endpoint `api/tree/getChildrenOfNode`.

Fixed

- Downloading metrics reports would fail due to timeout on large databases. Report CSVs are now streamed to the client as they are generated instead of being generated on the server and sent at the end.

- Social accounts would not properly be added to a space after accepting an email invite to join it.

- Fixed bug where extractors monitor will not print error, but just return 0 if queue is not found.

- Pagination controls are now vertically aligned and use unescaped ampersands.

- Changing the page size on dataset, collection, space listings would not properly update elements visible on the page. CATS-1030

- Added a max of 100 status messages on the page listing all extractions. Before this trying to list all extractions in the system was causing the JVM to run out of memory. CATS-1032

- Added padding to the top of the footer so that the superadmin notification does not cover buttons and the buttons at the end of forms are not too close to the footer and difficult to see.

1.7.4 - 2019-10-21

Fixed

- Extractors that don't specify EXTRACT as categories don't show up in the manual submission page. CATS-1023

1.7.3 - 2019-08-19

Fixed

- Fixed bug where metadata field names in the search box were being forced to lowercase, omitting search results due to case sensitivity.

1.7.2 - 2019-08-01

Fixed

- RabbitMQ plugin would throw an exception if rabbitmq server restarted. CATS-1012

Changed

- Changed internal regex wrapping syntax on search box queries to better handle complex terms.
- Updated core documentation, both the content and version of Sphinx. CATS-865

1.7.1 - 2019-07-09

Fixed

- Logging was accidently set to DEBUG, reverted it back to INFO

1.7.0 - 2019-07-08

This update will require a reindex of Elasticsearch. After deploying the update either call ``POST to /api/reindex`` or navigate to the ``Admin > Indexes`` menu and click on the ``Reindex`` button.

Fixed

- HTTP 500 error when posting new metadata.

Added

- Add archive button on file page which can trigger archive extractor to archive this file.

- Added S3ByteStorageService for storing uploaded bytes in S3-compatible buckets. CATS-992

- Added support for archiving files in Clowder and preparing an admin email if user attempts to download archived file. CATS-981

- Listen for heartbeat messages from extractors and update list of registered extractors based on extractor info received. For extractors using this method they will not need to manually register themselves through API to be listed. CATS-1004

- Added support for extractor categories that can be used for retrieving filtered lists of extrators by category.

Changed

- Improved Advanced Search UI to retain search results between navigations. CATS-1001

- Display more info on the manual submission page, link to ExtractorDetails view. CATS-959

- Clean up of Search pages. Renamed Advanced Search to Metadata Search. Added search form and Metadata Search link to main Search page. Consistent and improved search results on both search pages. CATS-994

- Updated the mongo-init docker image to aks for inputs if not specified as an environment variable. `docker run -ti --rm --network clowder_clowder clowder/mongo-init`

- Rework of the Elasticsearch index to include improved syntax and better documentation on the basic search page.

1.6.2 - 2019-05-23

Fixed

- Mimetype of RabbitMQ message body should be `application/json` instead of `application\json`. GH-12

1.6.1 - 2019-05-07

Fixed

- A double quote character in a metadata description disallowing edit of metadata definition. CATS-991

- About page should no longer show "0 Bytes", counts should be more accurate. CATS-779

- Fixed creation of standard vocabularies within a space.

- Slow load times in dataset page by removing queries for comments and tags on files within a dataset. CATS-999

- Send file delete events over RabbitMQ when a folder is deleted that contains files. CATS-995

Changed

- Improved the HTTP return codes for the generic error handlers in Clowder.

- Adjusted display of Advanced Search matching options to include (AND) / (OR). CATS-998

- Dataset page does not show comments on files within the dataset anymore.

- dataset-image previewer turned off by default since it is expensive for datasets with many files but does not much information to the dataset page.

- Removed unused queries for comments throughout the application.

Added

- Script to cleanup/migrate userpass account data to cilogon accounts.

1.6.0 - 2019-04-01

Added

- User API Keys are now sent over to extractors (instead of the global key). If user doesn't provide a user key with the request, one is gets created with name _extraction_key. If no user is available, the global key is used. CATS-901

- Ability to cancel a submission to the extraction bus. A cancel button is available in the list of extraction events. CATS-970

- Allow user to create and manage controlled vocabularies within Clowder.

- Cascade creation and deletion of global metadata definitions to all spaces. CATS-967

- New view for files and datasets offering a table view of the attached metadata.

- Add SUBMITTED event on the GUI of extractions and pass this submitted event id in extraction message. CATS-969

- Send email address of user who initiated an extraction so that extractors can notify user by email when job is done. CATS-963

- Extraction history for dataset extractors is now displayed on dataset page. CATS-796

- Script to verify / fix mongo uploads collection if file bytes are missing.

- Additional columns added to reporting API endpoint including date, parent resources, file location, size and ownership.

- Previewer for displaying internal contents of Zip Files. CATS-936

- Additional API endpoints for adding and retrieving file metadata in bulk. CATS-941

- Optional form to adding multiple metadata fields at once via UI under "Advanced." CATS-940

- CONTAINS operator added to Advanced Search interface and wildcards (e.g. ".*") now supported in search box. CATS-962

- New widget to add standard name mappings. BD-2321

- Add a new event for extractors "dataset.files.added" that is triggered when a user uploads multiple files at once via UI. CATS-973

- `/api/search` endpoint now supports additional flags including tag, field, datasetid, and others detailed in SwaggerAPI. CATS-968

- Add a dropdown to Advanced Search UI for filtering by space. CATS-985

Fixed

- Enhancements to reporting date and string formatting. Space listing on spaces report and on New Collections page now correctly return space list depending on user permissions even if instance is set to private.

- GeospatialViewer previewer added content to incorrect tab. CATS-946

- Handle 403 errors appropriately from the ZipFile Previewer. CATS-948

- Error when showing ordered list of tags and Elasticsearch included an empty tag. Also removed the ability to add empty tags both from the UI as well as the API. CATS-952

- In SuperAdmin mode, the Spaces page will correctly show all spaces. CATS-958

- In FileMetrics report, space and collection IDs are only added to the report once to avoid repeating.

- Apply 'max' restriction when fetching dataset file lists earlier, to avoid long load times for certain previewers. CATS-899

- Unable to edit metadata definition when description included newlines characters.

- Fixed user events not being created. Migrated to EventType enum class for tracking event types. CATS-961

- Loading indicator should now show on datasets page while files and folders are loading.

Changed

- Extraction events on File and Dataset pages are now grouped by extractor. The events view has been moved to a tab for both, and the File pages now have metadata and comments under tabs as well. CATS-942

- Cleaned up clowder init code docker image see README.

- Updated Sphinx dependencies in `doc/src/sphinx/requirements.txt` for building documentation.

1.5.2 - 2018-12-14

Fixed

- Filtering using Rabbitmq does not take into account exchanges. CATS-954

- The icon used for spaces in a few places was the wrong icon. CATS-955

- GeoJSON dataset previewer loads all files in a dataset. CATS-956

1.5.1 - 2018-11-07

Fixed

- Previewer tabs on the file page were showing default title "Preview" instead of the one defined in the previewer manifest. CATS-939

- Remove signup button if signup is disabled using `securesocial.registrationEnabled=false`. CATS-943

- Add flag smtp.mimicuser=false that will force emails to always come from the user defined in the configuration file instead of the Clowder user. CATS-944

1.5.0 - 2018-10-23

Warning: This update will reset all permissions assigned to roles. Please review the defintion of roles in your instance before and after the upgrade to make sure that they match your needs.

Added

- Ability to specify whether a previewer fires on a preview object in the database (`preview:  true`) or the raw file/metadata (`file:true`) in the previewer `package.json` file. CATS-934

- Support for adding multiple comma-separated tags on dataset and file pages.

- Ability to send events to extractors only if they are enabled in a space. Refactored some of the extraction code. Added more explicit fields to the extraction message regarding event type, source and target. Tried to keep backward compatibility. CATS-799

- Update Docker image's `custom.conf` to allow for override of Mongo and RabbitMQ URIs. BD-2181

- Script to add a service account directly into Mongo `scripts/create-account.sh`.

- Added a new view to display Extractor Details. CATS-892

- New API endpoints for proxying GET, POST, PUT, and DELETE requests through Clowder. There are still some issues with POST depending on the backend service (for example Geoserver). CATS-793 CATS-889 CATS-895

- Ability to enable disable extractors at the instance level (versus space level). CATS-891

- Add flag to specify not to run any extraction on uploaded files to dataset. By default, we always run extraction on uploaded files to dataset. BD-2191

- Tracking of view and download counts for Files, Datasets and Collections. CATS-374 CATS-375

- Ability to downloads CSV reports of usage metrics for Files, Datasets and Collections via new API endpoints. CATS-918

- Ability to provide API key in HTTP X-API-Key request header. CATS-919

- Extraction history for dataset extractors is now displayed on dataset page. CATS-796

- API route for creating a new folder now returns folder information on success.

- Offline updates for mongodb added to `scripts/updates`.

Changed

- If no local password and only 1 provider, redirect to the provider login page immediately. CATS-868

- Changing gravatar picture to be https in the database CATS-882

- Modified zenodo.json file to include more Orcid IDs. CATS-884

- Display more extractor information in each Space's "Update Extractors" view. CATS-890

- Clean up of the list of previewers page `/previewers/list` and added link to it from the admin menu. CATS-934

Fixed

- In a private mode, a superadmin can now see datasets in a space that he/she is not part of. CATS-881

- In private mode, users used to be able to see the list of spaces. Now they cannot. CATS-887

- In DatasetService, rename function of findByFileID to findByFileIdDirectlyContain. Add a new function find-ByFileIdAllContain to return back datasets directly and indirectly contain the given file. CATS-897

- Parameters subdocument is now properly escaped in rabbitmq message. CATS-905

- Removed erroneous occurrences of .{format} from swagger.yml. CATS-910

- Previews on the file page are now shown whether they are because of a `Preview` entry on the file added by an extractor or by `contentType` in `package.json` for each previewer. CATS-904

1.4.3 - 2018-09-26

Fixed

- File model not being deleted when it contains previews or sections. CATS-928

1.4.2 - 2018-08-21

Fixed

- Extractors printing the private key in error extraction status messages. CATS-903

1.4.1 - 2018-08-21

Fixed

- LDAP provider now properly sets identityId.userId. CATS-911

1.4.0 - 2018-05-04

Added

- Ability to disable username/password login provider. CATS-803

- Track original file name used when file was originally uploaded. SEAD-1173

- LDAP authentication. CATS-54

- Ability for users to create their own API keys. CATS-686

- Abilty for CiLogon provider to filter by LDAP groups.

- *exact* flag on collection and dataset API endpoints that accept a *title* flag. This will use exact matching on the title field instead of regular expression fuzzy matching.

- Having a temporary trash option. Can be set with useTrash boolean in the configuration file. CATS-780

- Track last time a user logged in.

- Add logic to check rabbitmq, mongo, clowder ready before creating default users with Docker compose. BD-2059

- Add Jupyter notebook examples of how to interacting with Clowder endpoints for file, dataset, collection and spaces manipulation.

- HTML previewer for text/html files. CATS-861

Changed

- File and dataset GET metadata endpoints now include their corresponding IDs and resource type information. CATS-718

- Cleanup of docker build process and how Clowder in launched in Docker. CATS-871

- Serving gravatar picture over https instead of http. CATS-882

- When the metadata.jsonld has a contextURL instead of a JsObject or JsArray show a popup with the link of the context instead of creating a link. CATS-842

- Changed permissions for the editor role CATS-921

Fixed

- Space admins can now delete metadata definitions. CATS-880

- Rolling log file wrote to wrong folder, now writes to logs folder.

- Now sends email when a user signs up using an external login provider. Due to this fix admins will receive an email a user logs on for the first time after this version is deployed when logging in with an external login provider. CATS-483

- Fixed dataset geospatial layer checkbox turn on/off and opacity. CATS-837

- Fixed GreenIndex previewer on clowder dataset page. BD-1912

- Only show the sort by dropdown in the collection page when the sort in memory flag is false. CATS-840

- Extraction status returns "Done" instead of "Processing" when one of the extractor fails CATS-719

- Avoid exception avoid exception in user events when unknown events don't match expected pattern (e.g. metadata events from another branch).

- Fixed bug where "show more results" would fail on Search. CATS-860

- Fixed bug where reindex of Elasticsearch would fail if extractors tried to index simultaneously. CATS-856

- Fixed bug of Account not active when using mongo-init to create user account. BD-2042

- Setting status for users on signup. CATS-864

- Person tracking previewer updated after changes to the associated metadata structure. CATS-730

- Hide incompatible extractors on `/datasets/:id/extractions` and `/files/:id/extractions` views. CATS-875

- Can now accept ToS even if account is not enabled. CATS-834

1.3.5 - 2018-02-23

Fixed

- Modifying subject for emails being sent when a user is added to a space. CATS-858

1.3.4 - 2018-02-05

Fixed

- Downloading datasets could take a long time before actual download started and could result in proxy timeouts. CATS-795

1.3.3 - 2017-12-21

Added

- Endpoint to get a list of traversing paths from datasets to the parent folders of the given file. CATS-811

- clowder.upload.previews flag to application.conf to turn on/off previews in upload page. CATS-813

Changed

- Send email with instructions when registerThroughAdmins=true. CATS-791

- Default showAll to true when listing spaces. CATS-815

- Move submit for extraction to the top on file page and dataset page. Remove parameter text field on Submit for Extraction page. CATS-794

- Add 'cat:' as prefix for typeOfAgent in UserAgent and ExtractorAgent constructors. Add filter or condition to check typeOfAgent is cat:extractor in getTechnicalMetadataJSON endpoint. CATS-798

Fixed

- Dataset geospatial previewer now has a max of 20 layers shown by default. The dataset page was taking too long to load for datasets with lots of files because of this. CATS-826
- Dataset descriptions of sufficient length no longer cause the page to freeze in tiles view.
- Tags lists now showing up to 10000 entries when using elasticsearch. Was defaulting to 10. SEAD-1169
- Add js route to get the JSONLD metadata of a file. GitHub-PR#2
- Geostreams POST /sensors lat and long are reversed. GLGVO-382
- Edit license breaks on names with apostrophes in them. CATS-820

1.3.2 - 2017-08-15

Fixed

- Elasticsearch searches are broken. CATS-783

1.3.1 - 2017-07-24

Fixed

- Upgraded Postgres driver to 42.1.1. Geostreams API was throwing an a "canceling statement due to user request" error for large datapoint queries with Postgresql versions 9.5+. CATS-771
- When doing a reindex all indices in elasticsearch were removed. CATS-772
- CILogin properly works by specifying bearer token in header.
- Collections id properly removed from child collections when deleting parent collection. CATS-774
- The modal for adding a relationship between sensors and datasets is now on top of the background and can be clicked. CATS-777

1.3.0 - 2017-06-20

Added

- Only show spaces, collections and datasets that are shared with other users under 'explore' tab. In application.conf, this is set by the *showOnlySharedInExplore* whose default value is false.
- Ability to download a collection. Download collection and dataset both use BagIt by default. CATS-571
- Ability to mention other users using '@' in a comment on a file or dataset. Mentioned users will receive a notification email and a notice in their event feed. SEAD-781
- Description field to metadata definition. SEAD-1101
- Improved documentation for the user interface.

Changed

- Ability to search datapoints, averages and trends using a start and end time.

- Ability to change how many items are displayed on the listing pages. SEAD-1149

- When downloading datasets there is no folder with the id for each file. SEAD-1038

- Datasets can be copied with *Download Files* and *View Dataset* permissions instead of just the owner. SEAD-1162

- Selections can now be downloaded, tagged or deleted directly from the top menu bar through the new action dropdown.

- Can assign any GeoJSON geometry to Geostreams entities in the PostGIS database, not just lat/long coordinates. CATS-643

- Attributes filter on datapoint GET endpoint can now include ':' to restrict to datapoints that match a specific value in their attributes. CATS-762

Fixed

- Binning on geostreaming api for hour and minutes. GEOD-886

- Returning the last average when semi is not selected and there is no binning by season.

- Removing space id from collections and datasets when the space is deleted. CATS-752

- Copy of dataset. When a dataset is copied, the newly created dataset will have the system generated metadata, previews, and thumbnails for the dataset and the files. CATS-729

- Return *409 Conflict* when submitting file for manual extraction and file is not marked as *PROCESSED*. CATS-754

- Listing of files in dataset breaks when user permissions in a space are set to View. CATS-767

- Reenabled byte counts on index and status pages.

- Miscellaneous bug fixes.

1.2.0 - 2017-03-24

Added

- Docker container to add normal/admin users for Clowder. BD-1167

- ORCID/other ID expansion - uses SEAD's PDT service to expand user ids entered as creator/contact metadata so they show as a name, link to profile, and email(if available). SEAD-1126

- Can add a list of creators to a Dataset and publication request(Staging Area plugin). This addition also supports type-in support for adding a creator by name, email, or ID, and adjusts the layout/labeling of the owner(was creator) field, and creator and descirption fields. SEAD-1071, SEAD-610

Changed

- Clowder now requires Java 8.

- Updated the POST endpoint `/api/extractors` to accept a list of extractor repositories (git, docker, svn, etc) instead of only one. BD-1253

- Changed default labels in Staging Area plugin, e.g. "Curation Objects" to "Publication Requests" and make them configurable. SEAD-1131

- Updated docker compose repositories from ncsa/* to clowder/*. CATS-734

- Improved handling of special characters and long descriptions for datasets and Staging Area publication requests SEAD-1143, CATS-692

- Default for clowder.diskStorage.path changed from /tmp/clowder to /home/clowder/data. CATS-748

Fixed

- Fixed email newsfeed template for new events, so that instances with malfunctioning email digest subscriptions can correctly generate digest emails. SEAD-1108

1.1.0 - 2017-01-18

Added

- Breadcrumbs at the top of the page. SEAD-1025

- Ability to submit datasets to specific extractors. CATS-697

- Ability to ask for just number of datapoints in query. GEOD-783

- Filter metadata on extractor ID. CATS-566

- Moved additional entries to conf/messages.xxx for internationalization and customization of labels by instance.

- *(Experimental)* Support for geostreams datapoints with parameters values organized by type. GLM-54

- Extraction messages are now sent with the RabbitMQ persistent flag turned on. CATS-714

- Pagination to listing of curation objects.

- Pagination to listing of public datasets.

Changed

- Only show Quicktime preview for Quicktime-VR videos.

- Organize public/published datasets into multiple tabs on project space page. SEAD-1036

- Changed RabbitMQ delivery mode to persistent. CATS-714

- Dataset and collection listing layout is not consistent with space listing layout.

Removed

- /delete-all endpoint.

Fixed

- Validation of JSON-LD when uploaded. CATS-438

- Files are no longer called blob when downloaded.

- Corrected association of JSON-LD metadata and user when added through API.

- Ability to add specific metadata to a space. SEAD-1133, SEAD-1134

- Metadata context popups now always properly disappear on mouse out.

- User metadata @context properly filled to required mappings. CATS-717

1.0.0 - 2016-12-07

First official release of Clowder.

1.3 License

Clowder is licensed under the NCSA license. Please see below for the full text (*Clowder License*). Please do not modify the conditions and make the license available in any derivative work. You must not use the names of the authors to promote derivatives of the software without written consent.

1.3.1 Contributors

Following is a list of contributors in alphabetical order:

- Aaraj Habib
- Ashwini Vaidya
- Avinash Kumar
- Ben Galewsky
- Bing Zhang
- Brock Angelo
- Chen Wang
- Chris Navarro
- Chrysovalantis Constantinou
- Constantinos Sophocleous
- Dipannita Dey
- Gene Roeder
- Gregory Jansen
- Indira Gutierrez

- Inna Zharnitsky
- Jim Myers
- Jong Lee
- Kastan Day
- Kaveh Karimi-Asli
- Kenton McHenry
- Luigi Marini
- Maria-Spyridoula Tzima
- Mario Felarca
- Max Burnette
- Michal Ondrejcek
- Michael Johnson
- Michelle Pitcel
- Mike Bobak
- Mike Lambert
- Nicholas Tenczar
- Nishant Nayudu
- Peter Groves
- Rob Kooper
- Rui Liu
- Sandeep Puthanveetil Satheesan
- Smruti Padhy
- Theerasit Issaranon
- Tim Yardley
- Todd Nicholson
- Varun Kethineedi
- Ward Poelmans
- Will Hennessy
- Winston Jansz
- Xiaocheng Yuan
- Yan Zhao
- Yibo Guo

1.3.2 Clowder License

Copyright (c) 2013 University of Illinois at Urbana-Champaign All rights reserved.

Developed by: Image and Spatial Data Analysis Division (ISDA) National Center for Supercomputing Applications (NCSA) University of Illinois at Urbana-Champaign (UIUC) http://isda.ncsa.illinois.edu/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

Neither the names of ISDA, NCSA, UIUC, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

1.4 User Guide

This user guide is intended for first time users. It covers some basic initial interactions with the system through the user interface.

1.4.1 Signing Up

There are 2 ways to login to Clowder. Through a third party application or a local account.

Signing up with a Third Party App

The first one is to use a third-party network, like google, crowd, facebook, twitter, linkedin. They are enabled in some instances on Clowder, and you can ask your Clowder administrator to enable them if they are not.

Each Clowder instance manages its own user database, for example: https://clowderframework.org/clowder/.

If you want to login using this method, you can click on the Login button in the top-right corner as marked by a blue square on Figure 1.
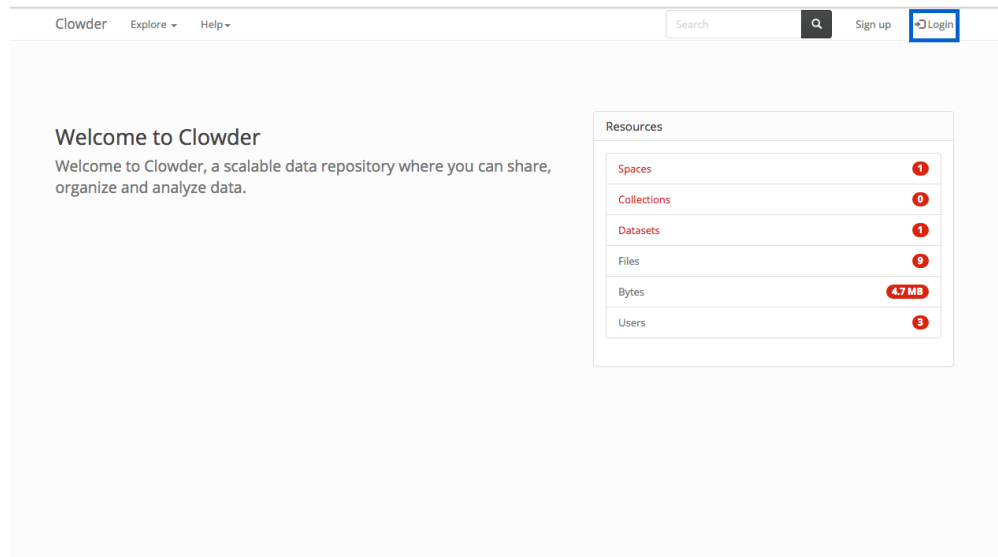
Login

After that, you can click on the icon of the third party app you want to login with. I will use google as an example, but the process is similar for all the 3rd party libraries.
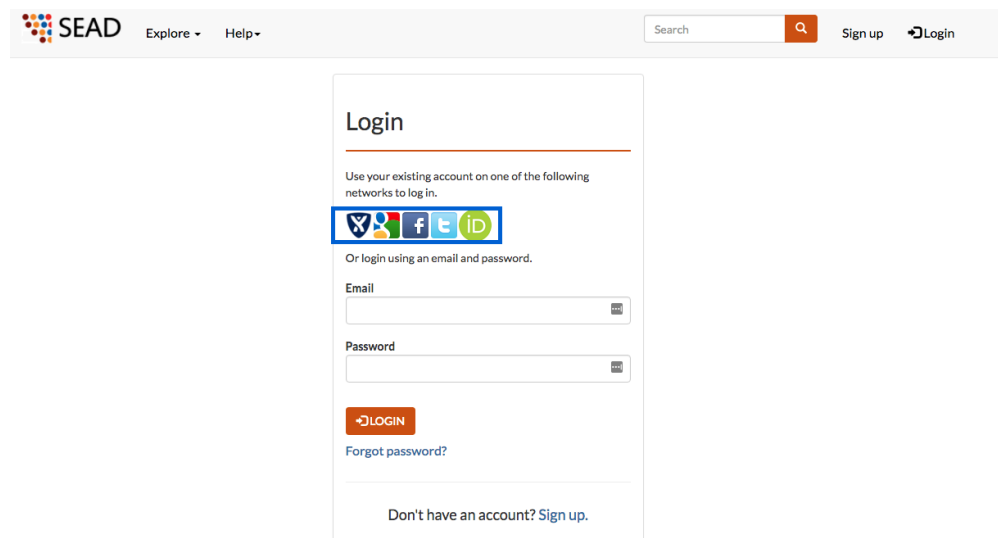
After clicking on the Google link, it will ask you to login for your google account, and on the first time it will ask you for permissions to get your name and email from the google account.
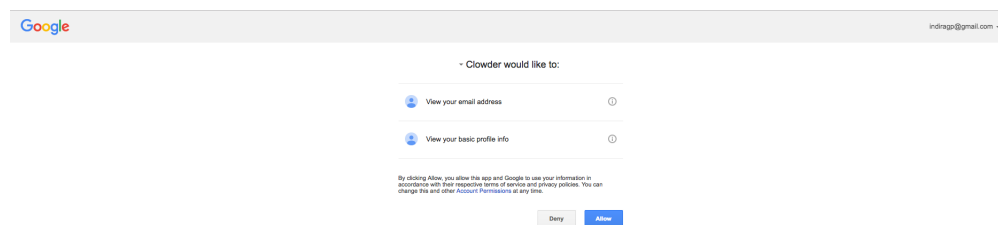
After you click on the Allow button, you will be redirected to the Terms of Service page. Which you must accept before getting access to Clowder.

After accepting the terms of service, you will be redirected to the Clowder Home Page.

Signing up through a Local Account

Click on the highlighted link in blue in your Clowder instance (https://clowderframework.org/clowder/r)
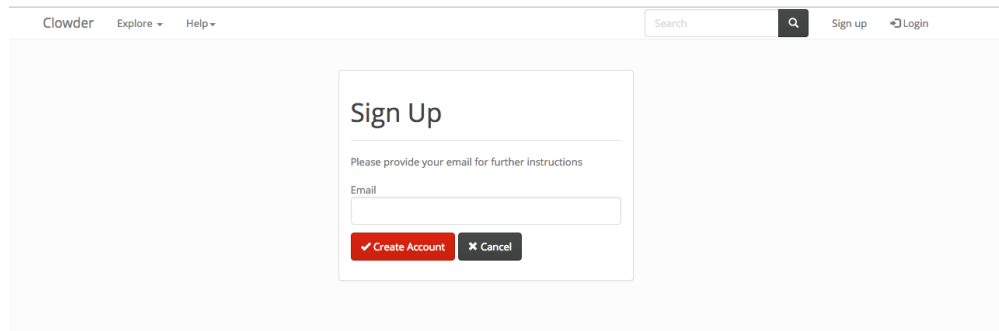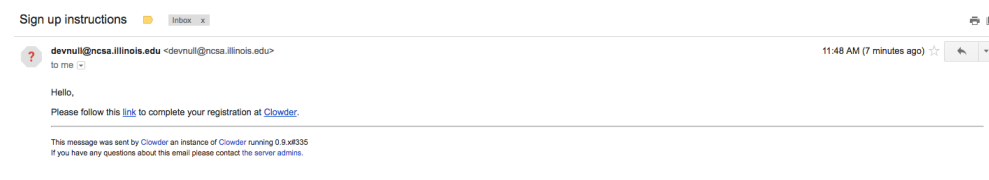
It will ask you for your email
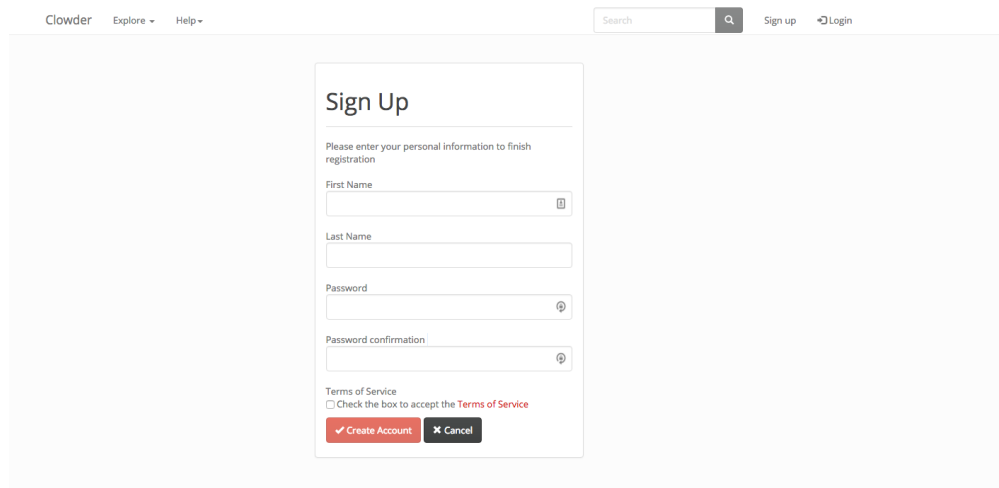
After you input your email and click on create an account, you will receive an email with instructions to continue the signup process

After clicking on the link you will get redirected to the sign up page

After input your information, and desired password you can login with your email and password used in the step above

And you can login to Clowder
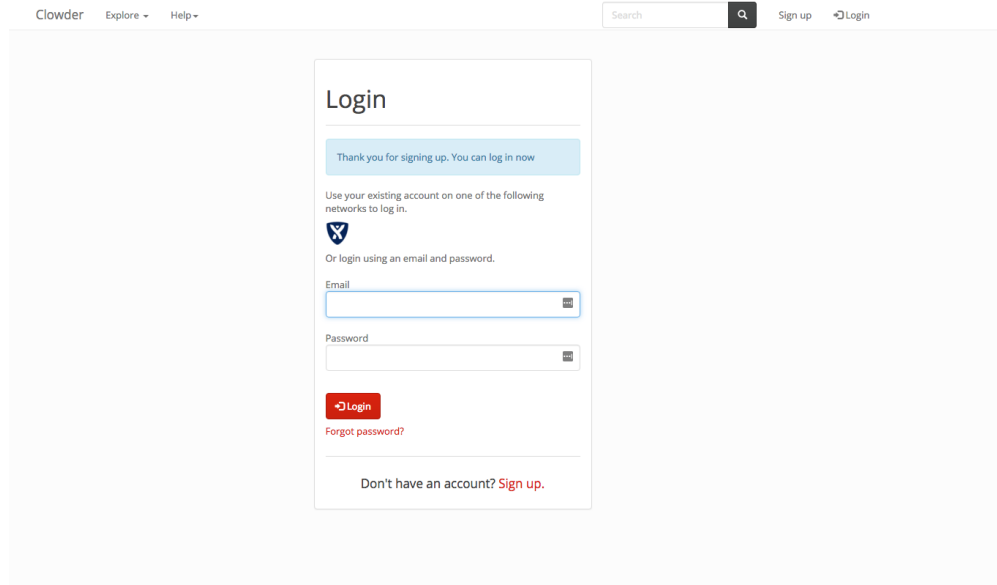
Note: Some instances require users to be approved before being able to use Clowder.

1.4.2 Home Page

The home page is where the site redirects you after logging in. You can navigate back to your home page by clicking on the Clowder link in the top navigation (Left to the you box highlighted in blue).

In Clowder, there are 3 main entities for organizing files. They are spaces, datasets, and collections. In your home page, you have easy access to creating new spaces datasets and collections. The same functionality is available anywhere in the application through the top navigation 'Create' dropdown. Both of these are highlighted in green squares on the image below.

You can also see the spaces, datasets and collections that you have created by clicking on the links for the tabs in blue. The same functionality is available in the 'You' dropdown. Also highlighted in blue.

In the home page, you can also go to your profile page by clicking the profile button to the left of the create buttons highlighted in green.

The activity tab displays events on datasets, collections, spaces, files and users that you follow. It displays when someone adds a comment to a file, a dataset to a space, a file to a dataset when you are added and removed from spaces, among others.

In the top navigation, you can also access the listing of all spaces, dataset, collections and users through the 'Explore' dropdown, indicated by a purple square.
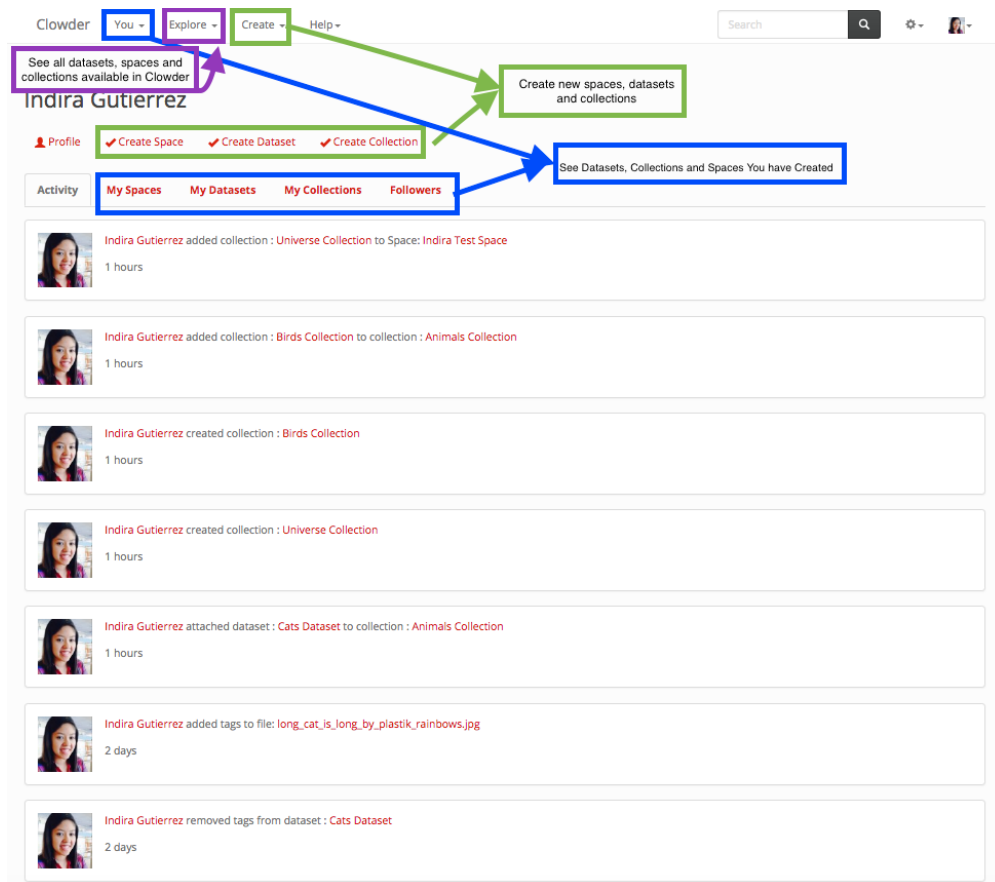
1.4.3 Datasets

A group of files that through some defined relationship or corresponding metadata are strongly tied together and not representable otherwise by the individual files.

Creating Datasets

Datasets can be created from the home page, the dataset list page, the top navigation, within a space and within a collection To create a dataset from the home page select the Create Dataset button displayed in a blue box in the next figure.
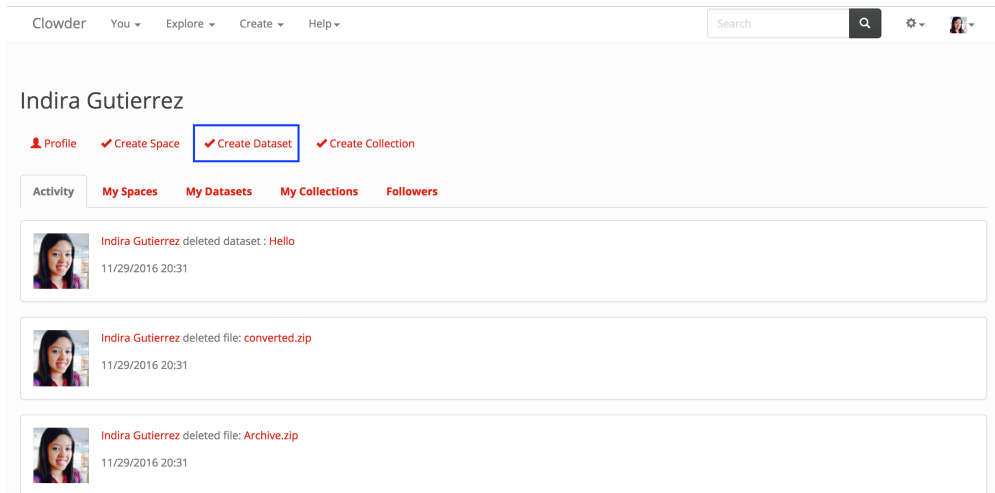
To use the create dialog in the top navigation. First click in the green dropdown button shown in blue in the next figure, and then select Datasets in green in the next figure.
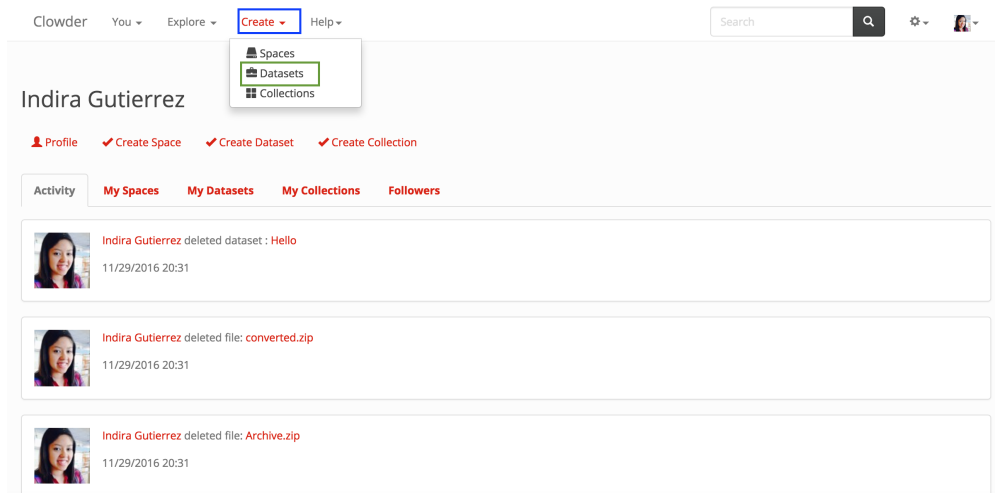
To create within the dataset list page. Click on the Create button in the top right of the page. (The list datasets page is accessed by clicking on Explore > Datasets)
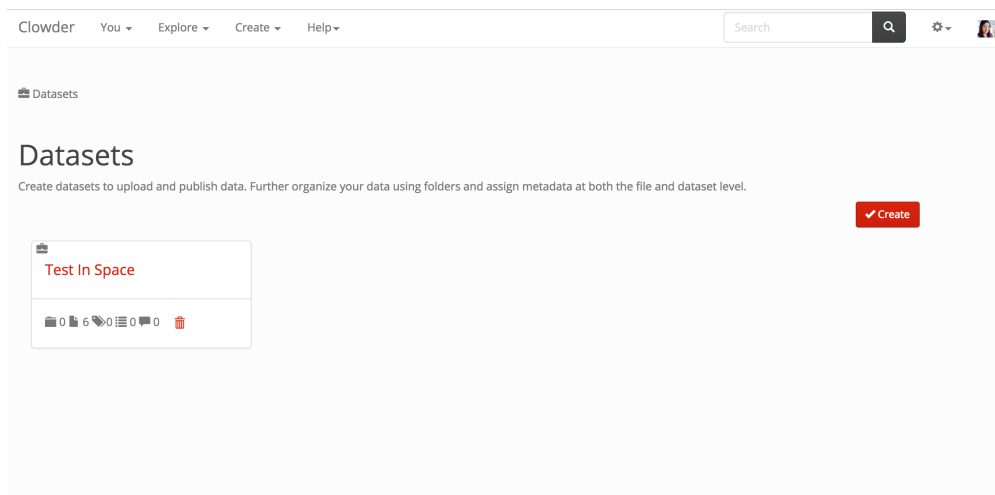
To create a dataset within a space. Go into a space page and selected the Create dataset button displayed in a blue box in the next picture.
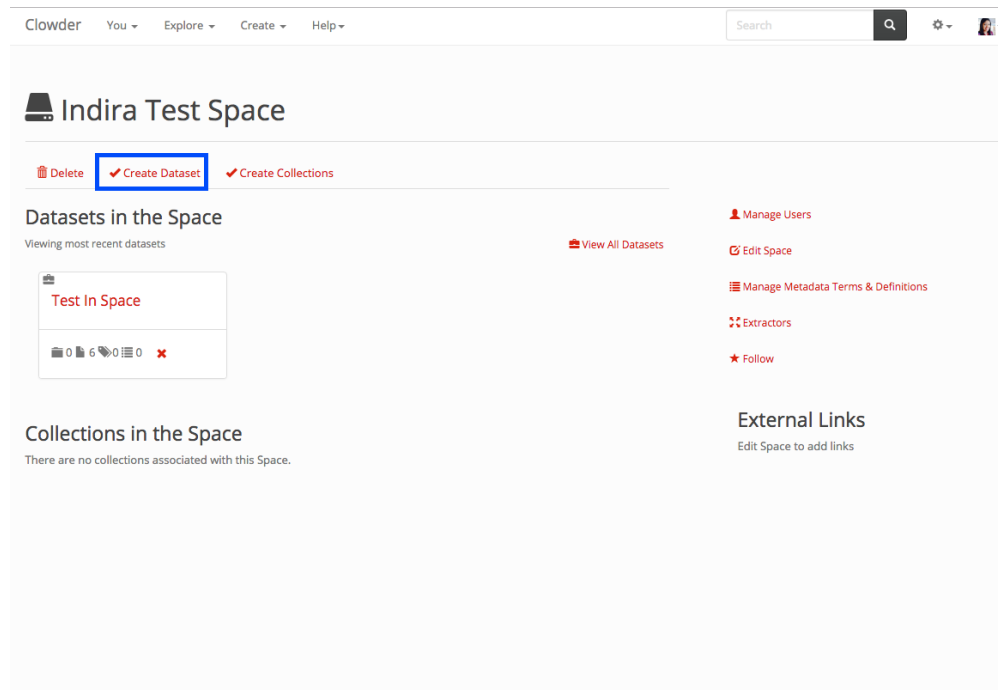
To Create a dataset within a collection. Go into a collection page and select the create button displayed in a blue box in the next picture.
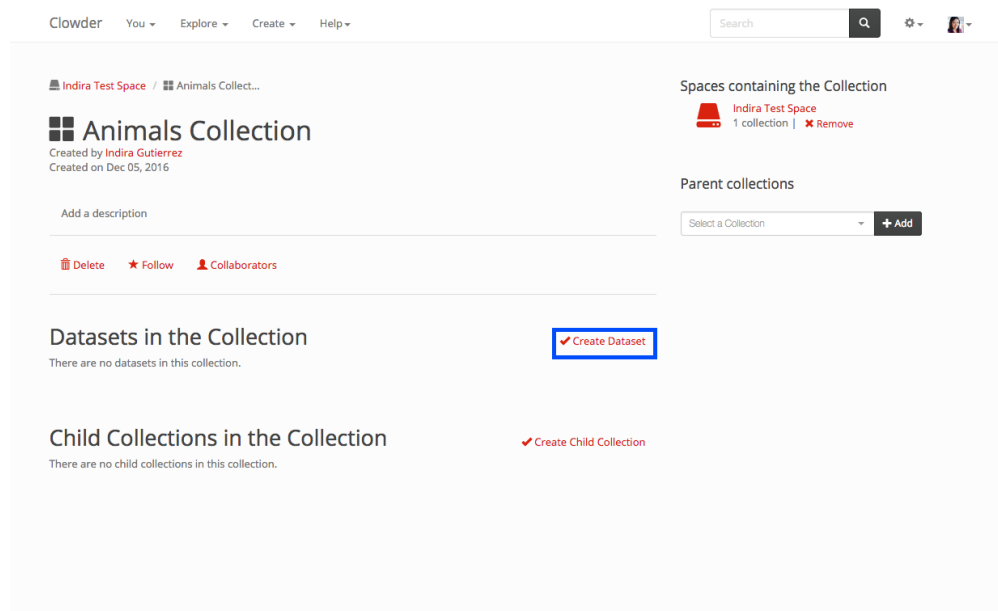
After selecting either of the above methods to create a dataset. The picture below shows up to create the dataset. The dataset only requires a name. You can optionally select a space you want to share the dataset with. If you start the creating process within a space. The space will be preselected, and can be changed or removed. To Create the dataset click on the create button in the bottom of the page

The create dataset is a 2 step process. After adding a name, you can add files to the dataset. This can be done by dragging files to the interface and then selecting the Upload button (pointed by the blue arrow). Or you can click on the Select files button (pointed by the green arrow) which opens a navigation page within your system to select the files you want to upload, and then clicking the upload button.

Editing a Dataset

In a dataset you can edit the name, description and license. In order to edit the dataset name, hover over the name and an edit button will show up as displayed in the next image surrounded by a blue box.
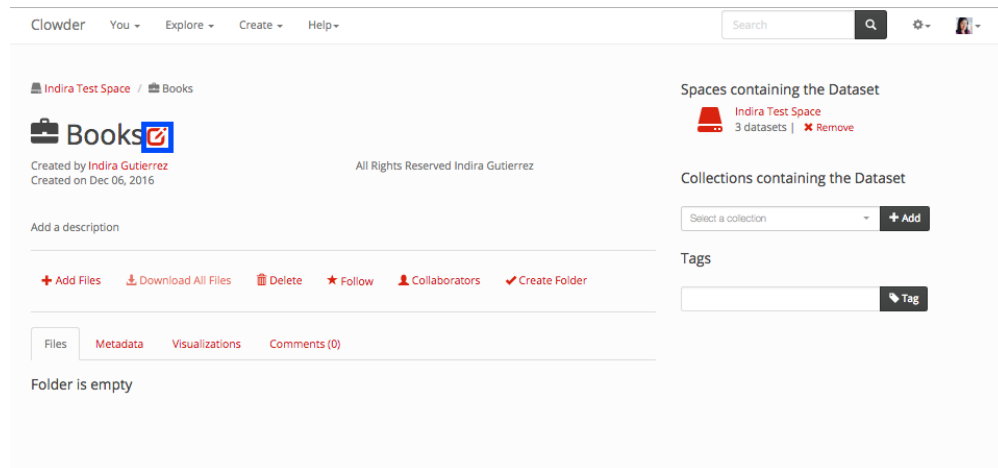
After clicking on the edit button, an input field pre filled with the current name is displayed, you can edit or cancel the name update. The next image shows the input field and the buttons that show up for changing the dataset name
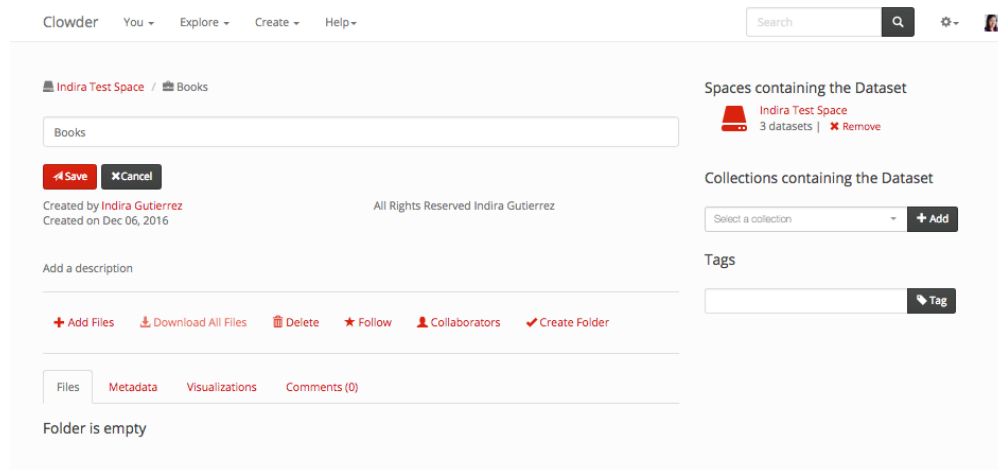
A similar process is used for updating the description and license, by hovering over each of those sections.

Adding Files

You can also add files after creating a dataset. Just click on the Add Files button displayed in the picture below within a blue box.
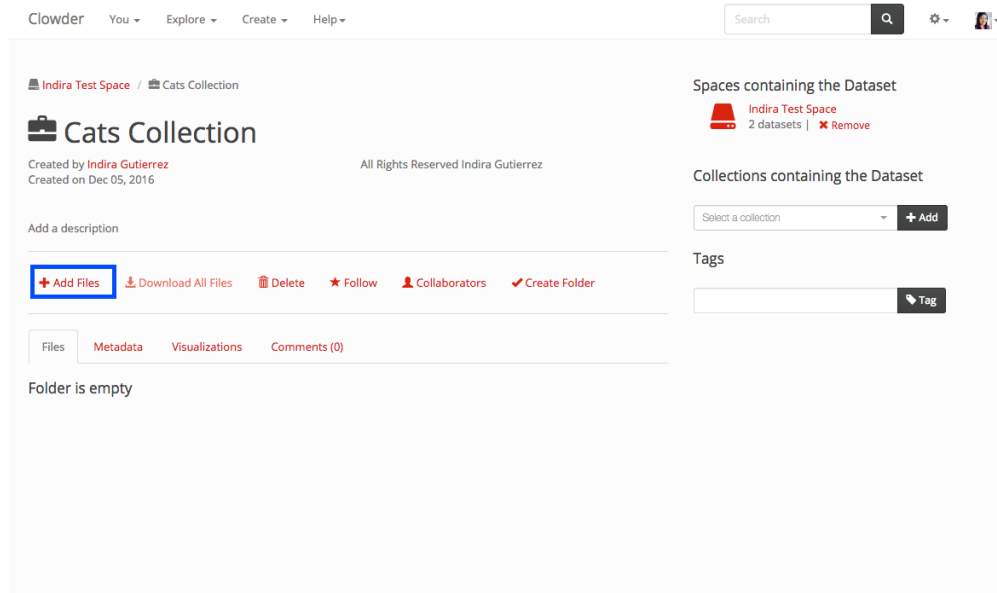
After that a page similar to the 2nd step of creating a dataset is presented. You can drag files or use the 'Select Files' button to look for files in your machine. After that click on the Upload button to upload your files. You can go back to the space page by clicking the left arrow next to the title. The dataset link with the dataset name below the title or in the breadcrumbs. The three ways for going back to the dataset are surrounded by blue boxes in the image below.
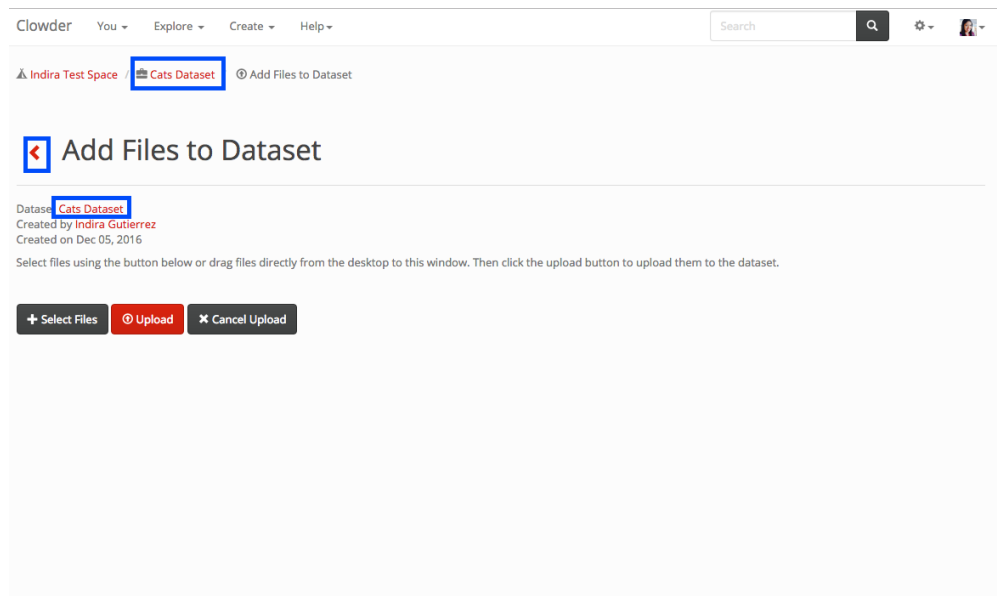
Editing a File Name

Similarly to a dataset, a file's name, description and license can be edited. In order to do so, hover over the field you want to update. Below the icon that is next to the description is displayed within a blue box
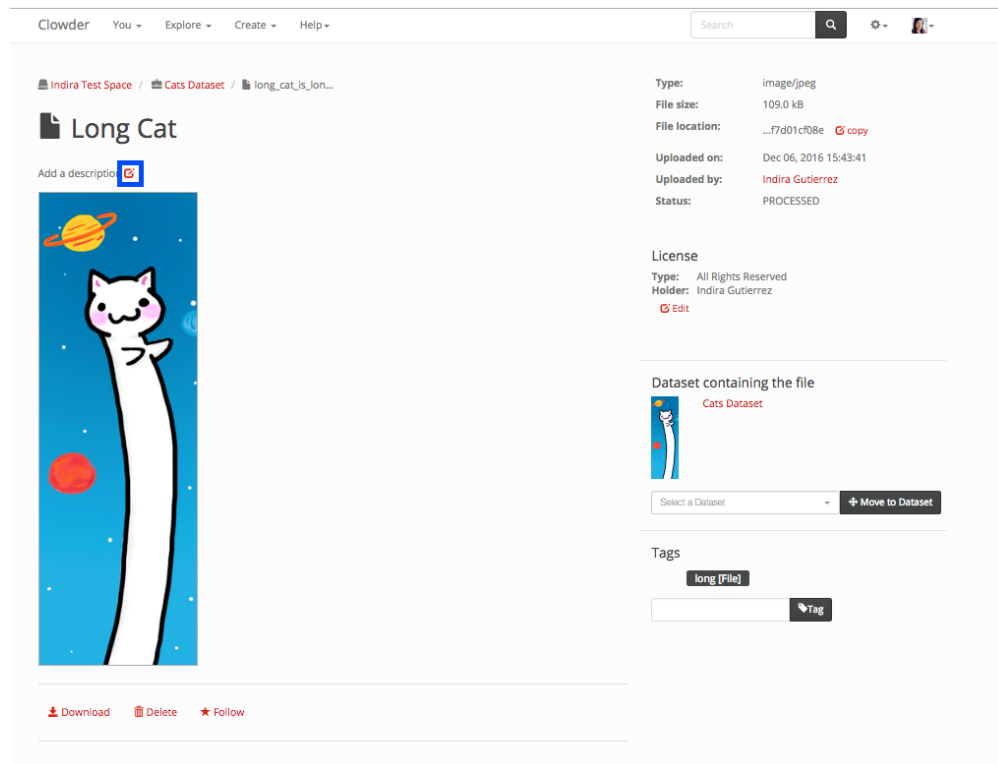
Then an input field and Save and Cancel buttons show up where you can edit the description if one exist, or add one if none exists.
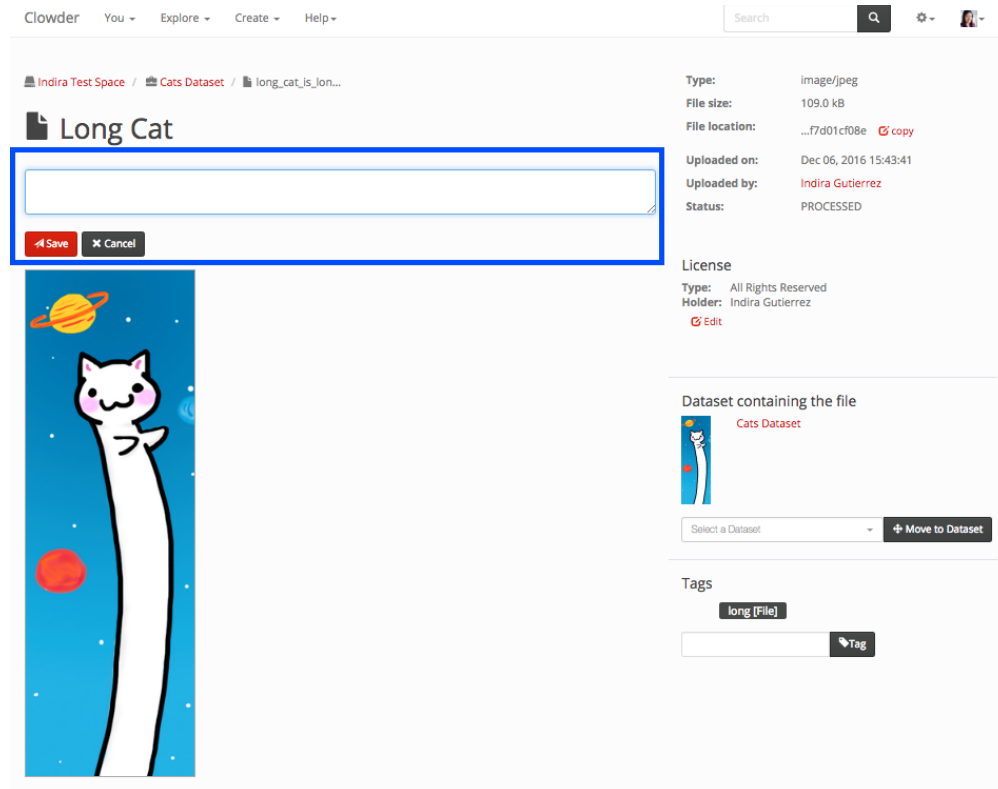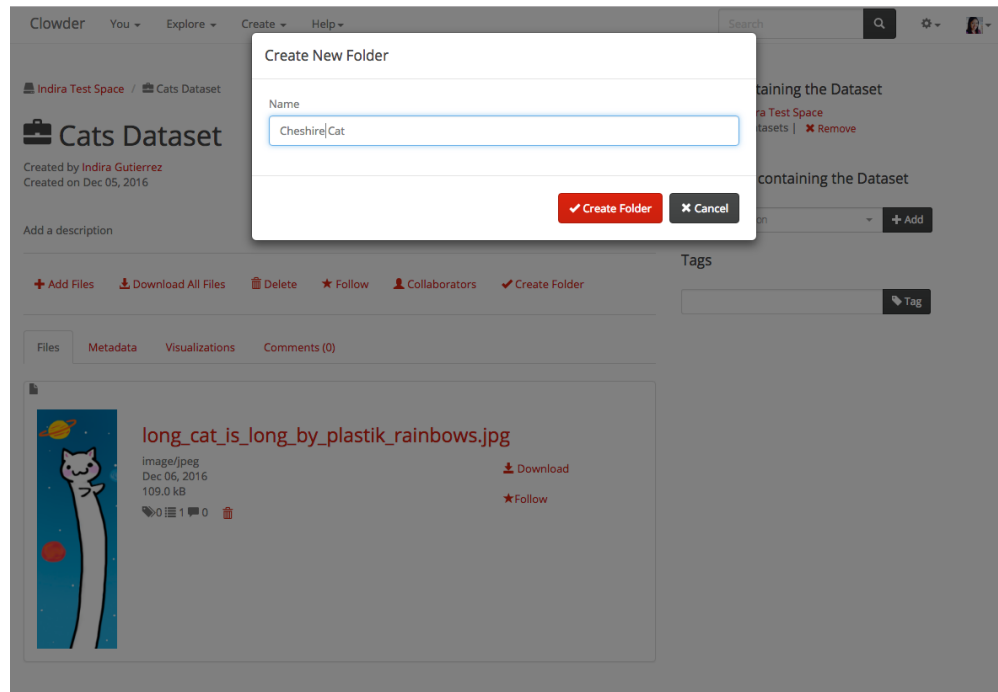
Creating Folders

To create a folder within a dataset, you need to go into a dataset page and click on the Create Folder button displayed within a blue box in the next image.
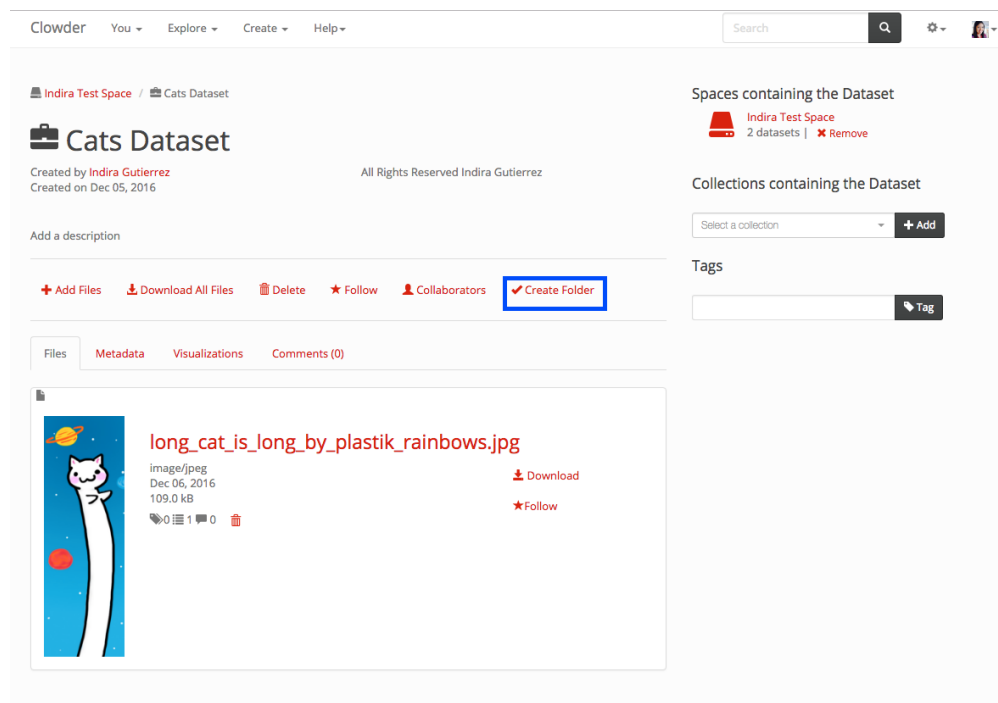
After clicking on the Create Folder button, a popup appears where you can input the name and click again on Create Folder button.


You can add files to a folder by clicking on the folder name, and then clicking on the Add files button as indicated above when adding files to a dataset.

Editing a Folder Name

A folder name can be changed by hovering over the folder name an edit icon shows up. The icon is displayed on the next image within a blue box.
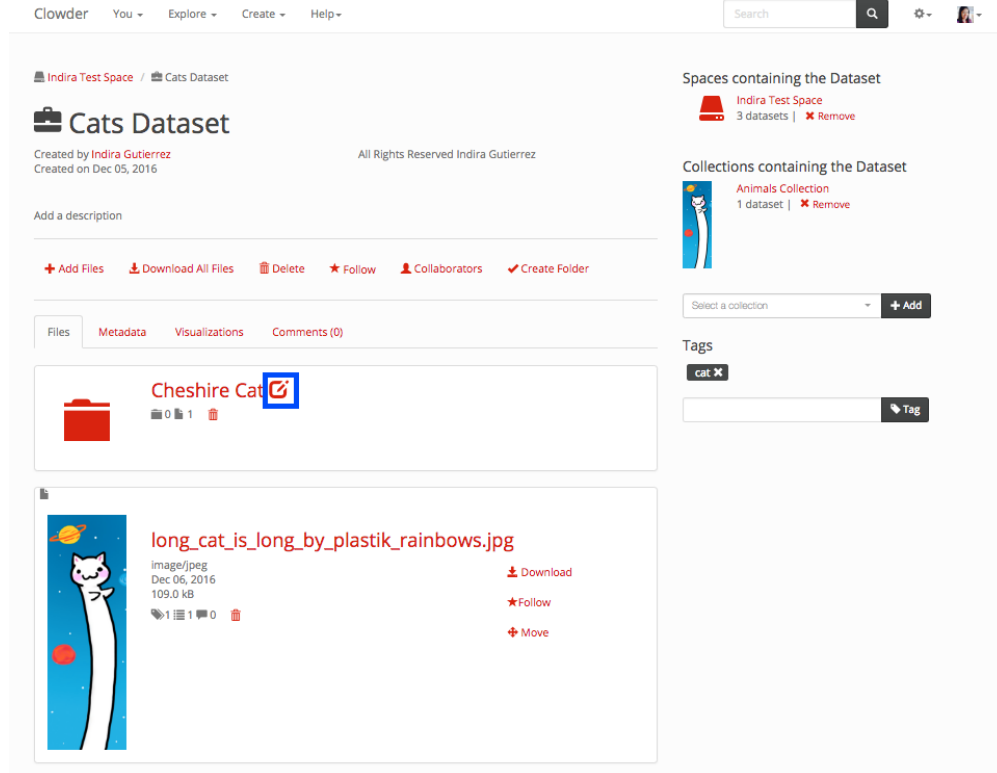
After clicking on the button the folder tile updates to have an input field pre filled with the current folder name and you can change it or cancel the name change.
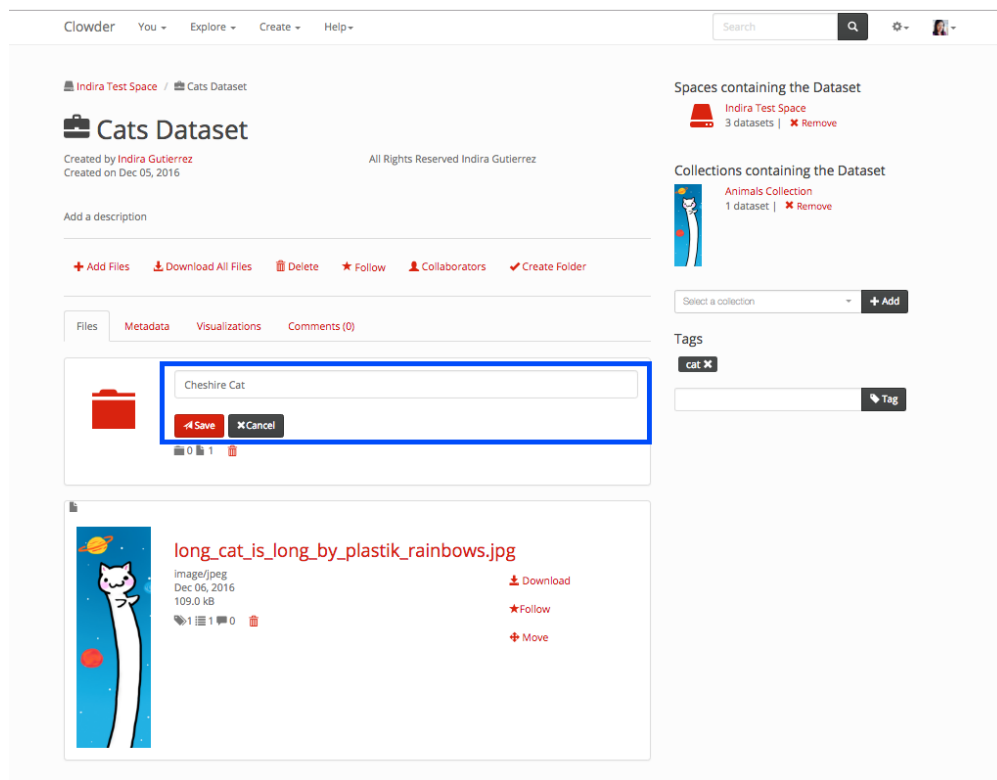
Moving Files

A file can be moved to other folders in the dataset, or between datasets.

To move a file within the dataset (to other folder) you click on the Move button in the file tile as indicated in the image below by the blue box. A popup appears with the list of available folders the file can be moved to
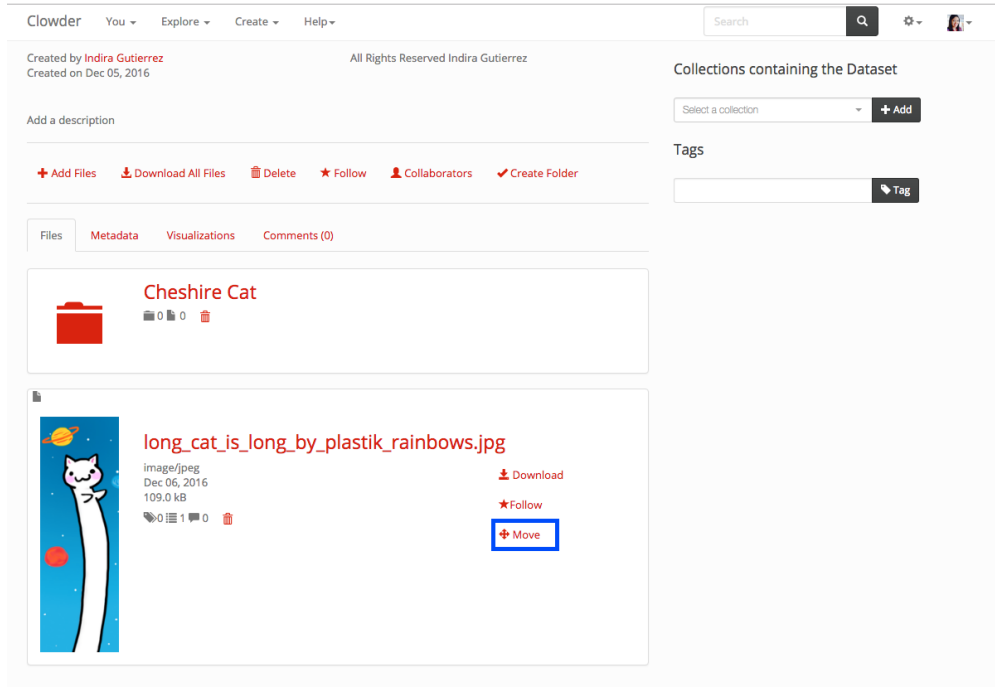

To move a file between datasets, you have to click on the file name to go into the file page. And on the right navigation click on the dropdown in the Datasets Containing the file section, select the dataset you want to move the file to and click on the 'Move to Dataset' button. The section with the dropdown and button is shown within a blue box in the next image.
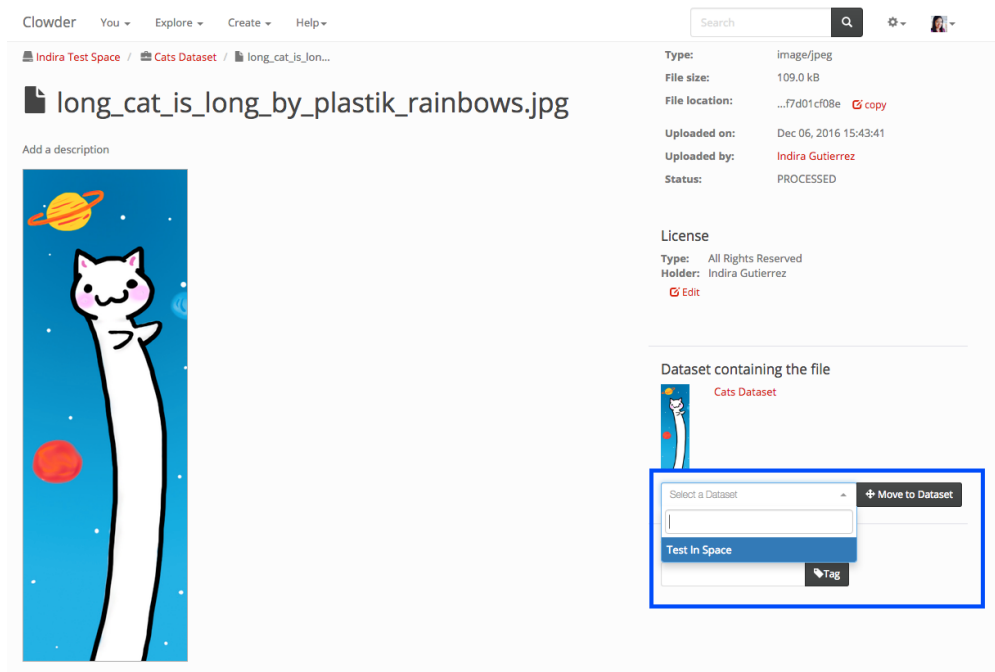
Adding to a Space

A dataset can be added to a space at the creation of the space. Or it can be added afterwards. For adding a dataset after it has been created. Click on the dropdown in the Spaces containing the dataset section. And a list of the spaces that you have the 'Add dataset to Space' Permission shows up. Select the space you want to add the dataset to and click on the Add button next to the selected space. The section with the dropdown and the button is displayed within a blue box in the next image.
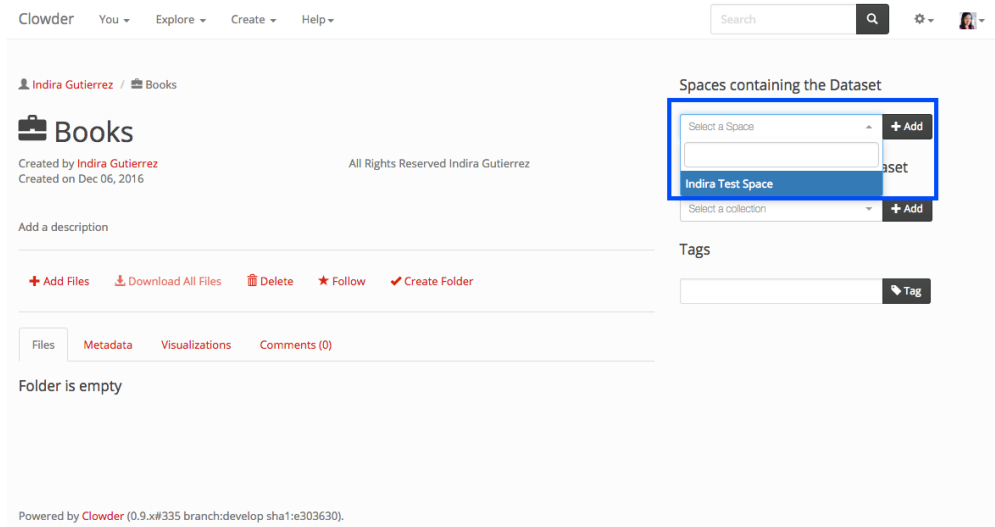

Removing a Dataset from a Space

To remove a dataset from a space you can do it from the space page or the dataset page.

To remove it from the dataset page click on the remove button in the 'Spaces containing the dataset section'. The button is displayed in a blue box in the next image.
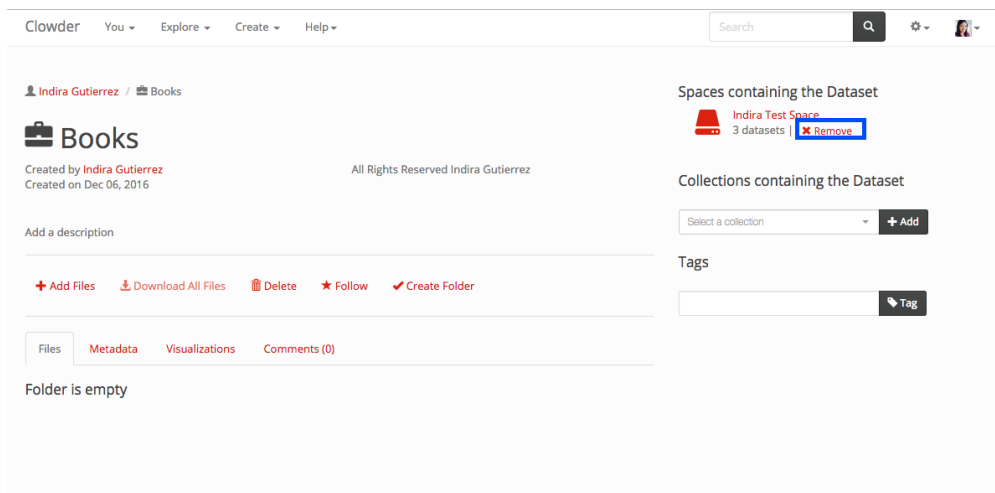
Within a space page in the tile for the dataset you want to remove. You can click on the x button to remove it from the space. Note: This does not delete the dataset from Clowder. The x's locations are marked by a blue box on the next image.
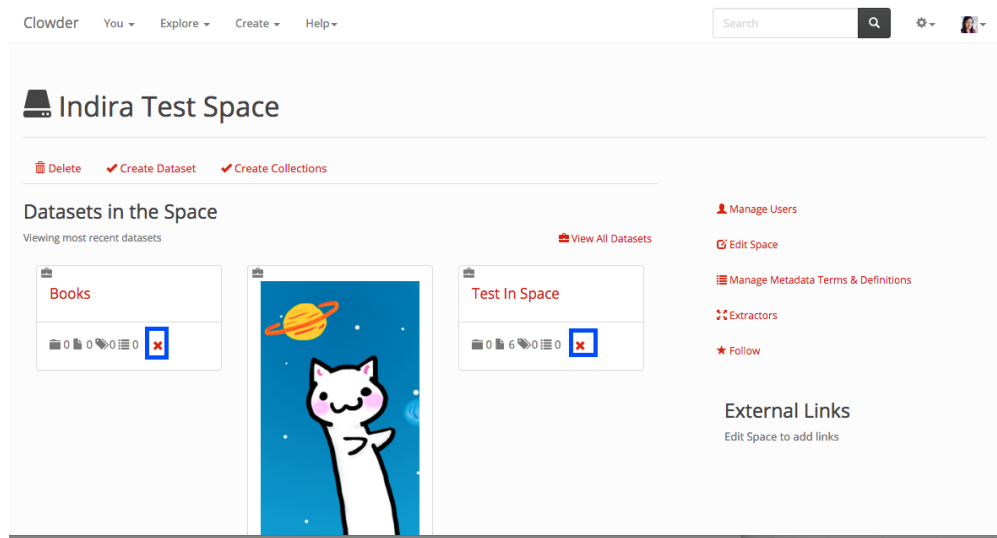
In both of the aforementioned ways there is a popup to confirm that you want to remove the dataset of the space. Where you can cancel or remove the dataset from the space.

Adding Metadata

Metadata is simply data about data. Metadata can be added to datasets or individual files.

To add metadata on a dataset click on the metadata tab, indicated by a blue box in the next image. And then click on the dropdown with the available metadata definitions and input the necessary data. The metadata drop down is indicated by a green box in the image below.
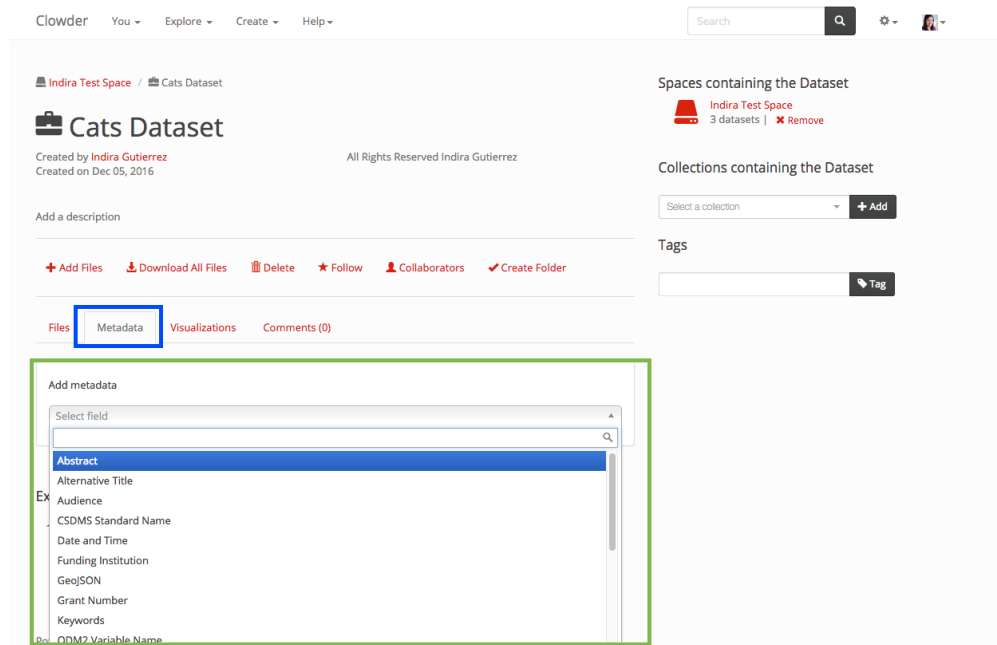

To add metadata to a file. In the file page below the previews section a dropdown similar to the dataset one appears. It is displayed in a blue box in the next image.
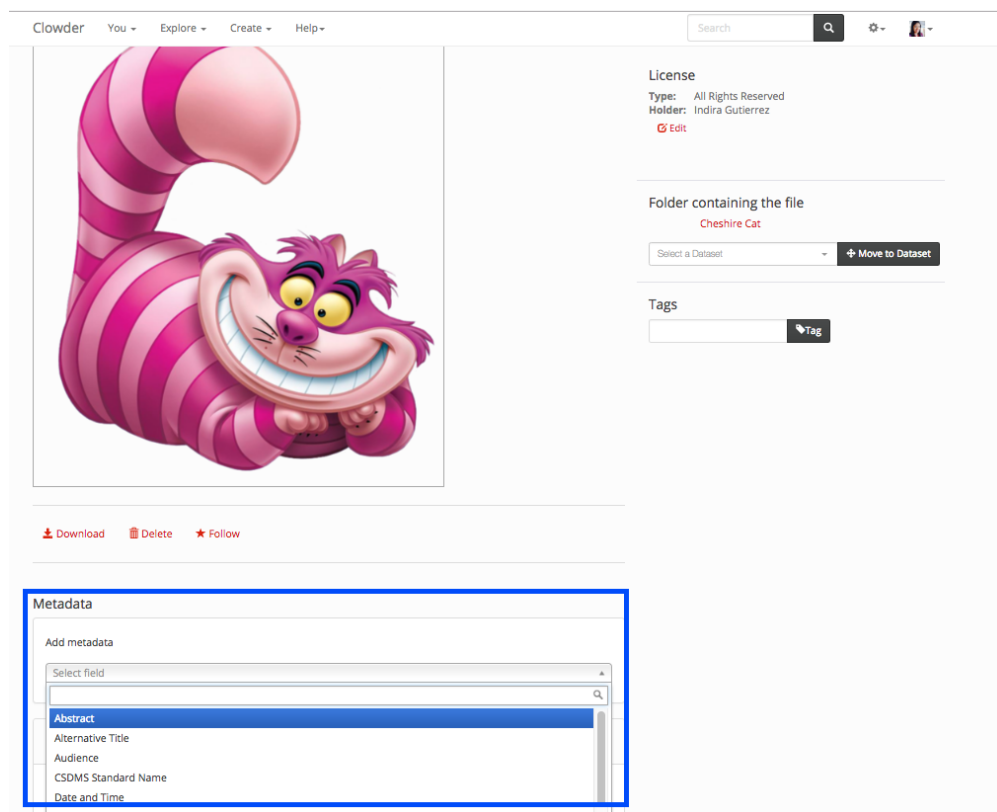
Adding Tags

Tags are a short string, e.g. one or two words, associated with a file or data set used to categorize or index its contents. To add a tag to a dataset write the tag in the input box in he tags section in the right navigation. To save it press the enter key or click on the tag button next to the input. The tag section is highlighted with a blue box in the next image.
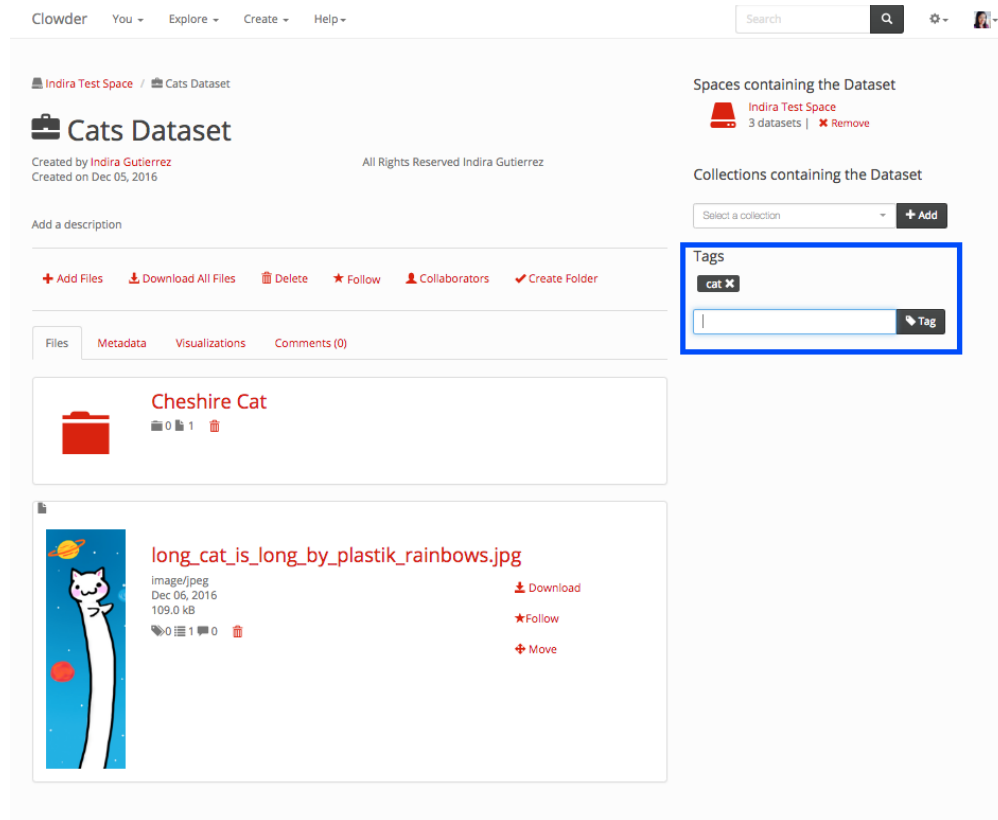
To add tags to files input the tag name in the tag section in the right navigation. To save it press the enter key or click on the tag button next to the input. The tag section is highlighted with a blue box in the next image.
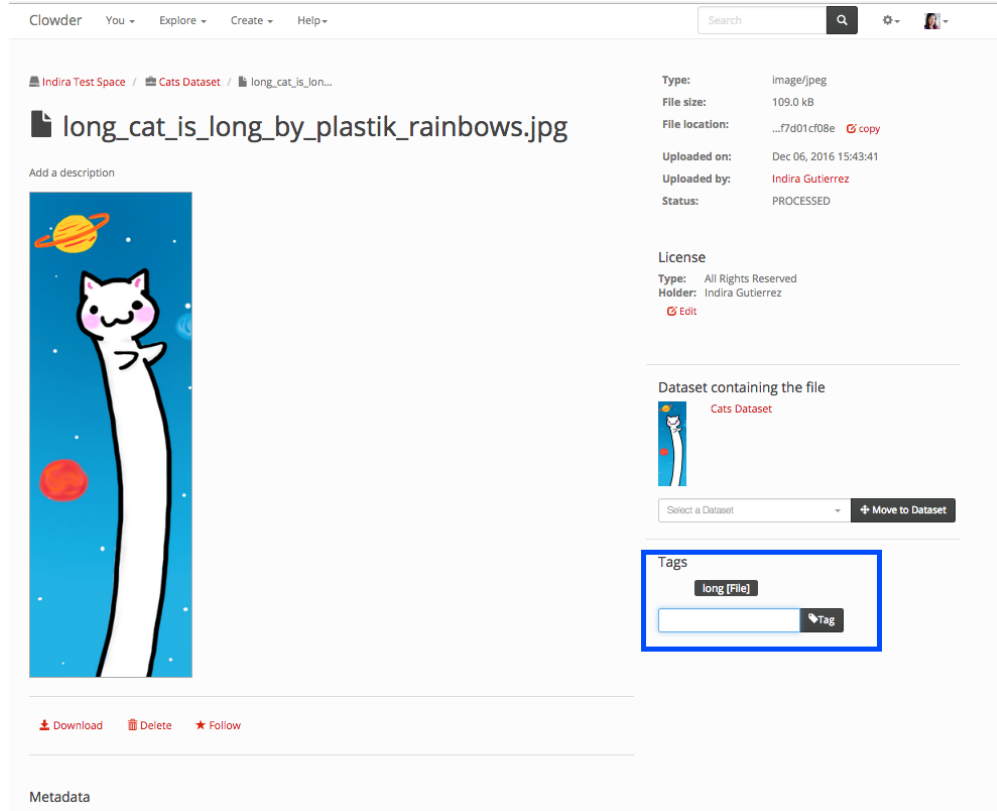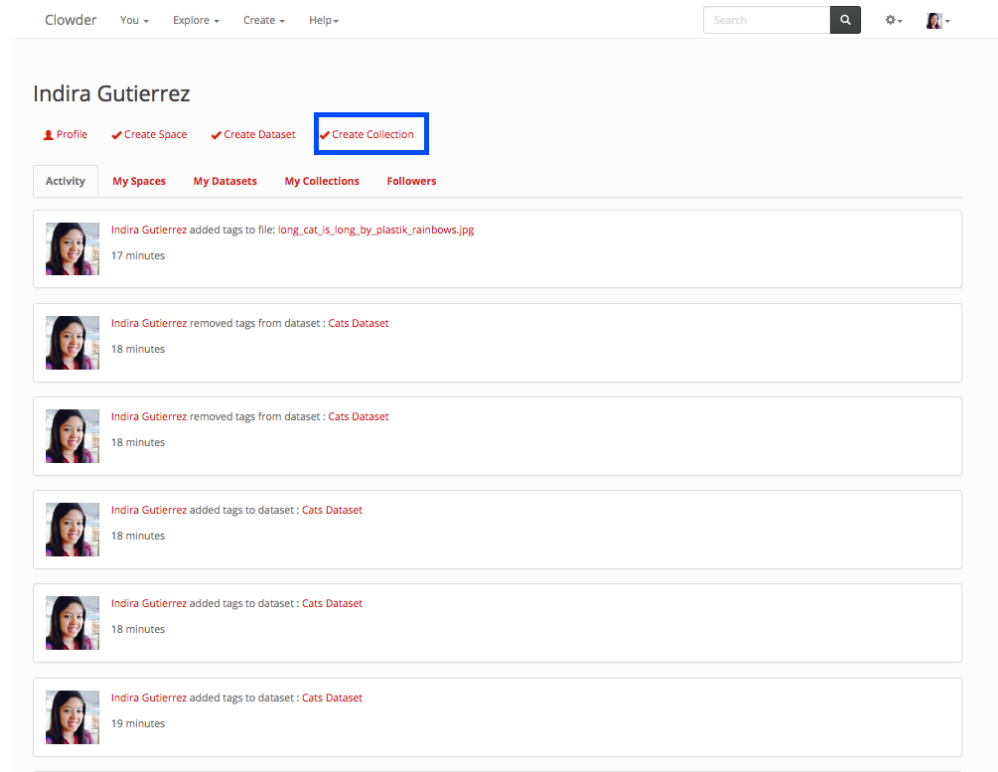
1.4.4 Collections

Collections are a user defined group of datasets and other collections.

Creating a Collection

Similarly as a dataset, a collection can be created from the home page, the top navigation page, within the collection list or within a space.

To create a collection from the home page click on the 'Create Collection' button below your name in the links section. Displayed in blue in the next image.
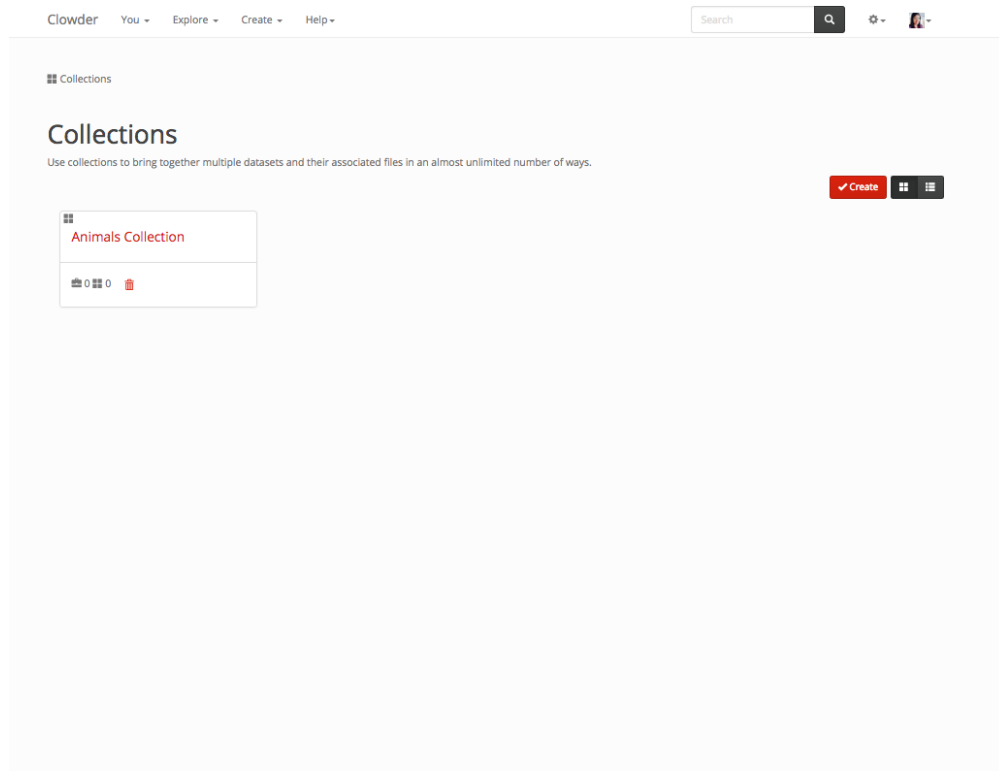
To create a collection from the top navigation. Click in the create dropdown in the top navigation displayed in blue in the next image. Next click in collections displayed in green in the next image.
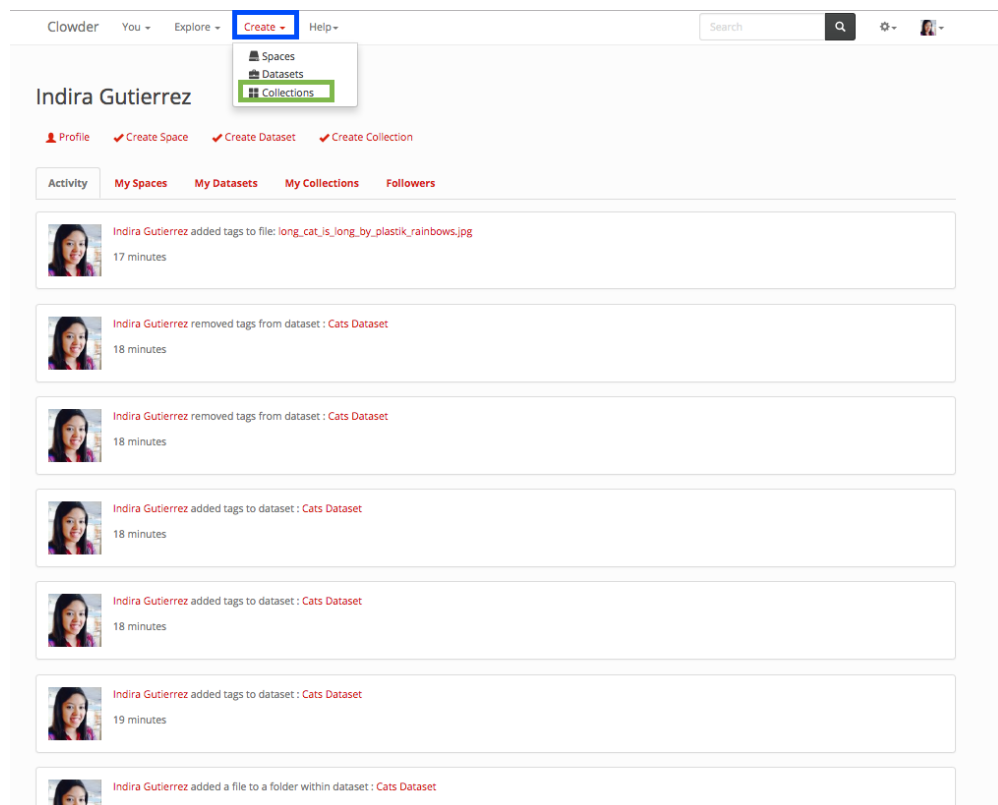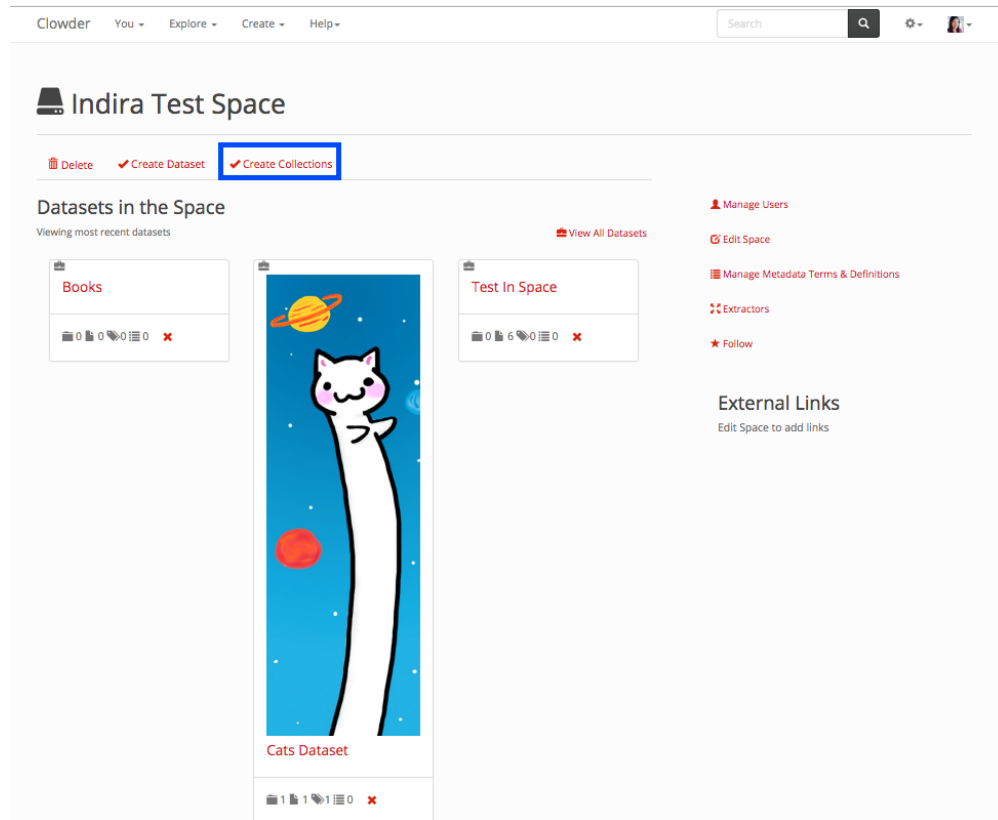

To create a collection from the list of collections page click on the create button in the top right. The list of collections

is displayed below.

To create a collection within a space, go into the space and click on the create collection button displayed in the image below within a blue box.

A collection has a name, description and space. The name field is the only required field. After you start the creating process for a collection with any of the above methods. A page like the one in the image is below. Once you have input the information that you want for your space. You can click on the create collection button at the end of the page, and you will be redirected to your new collection page.

Editing a Collection

In a collection, you can edit the name and the description. In order to do so, hover over the collection name or description and an edit symbol will show up. The button that shows up is displayed within a blue box on the next image.
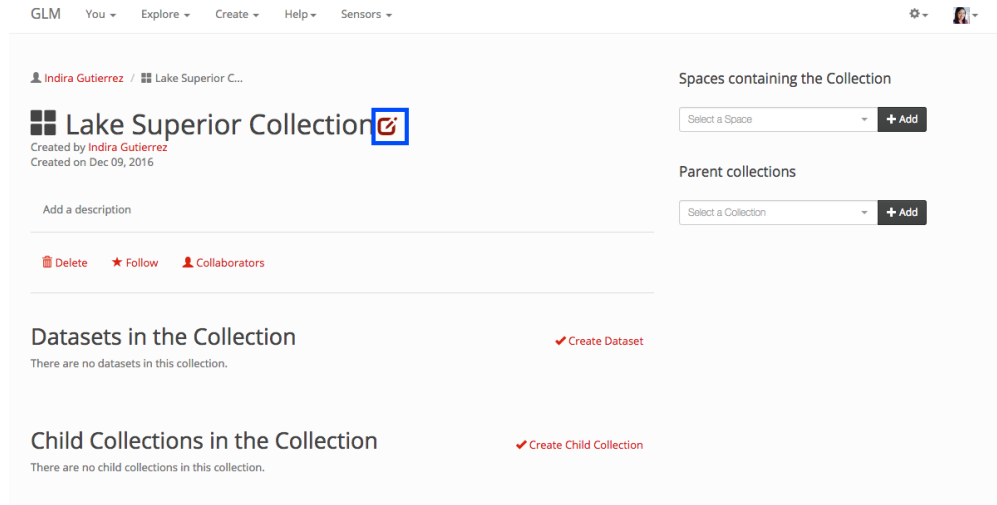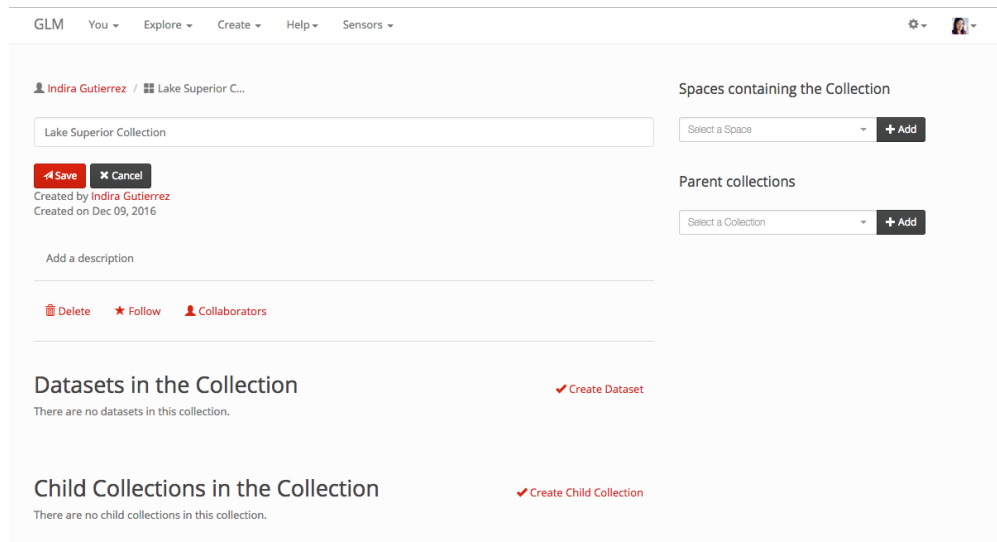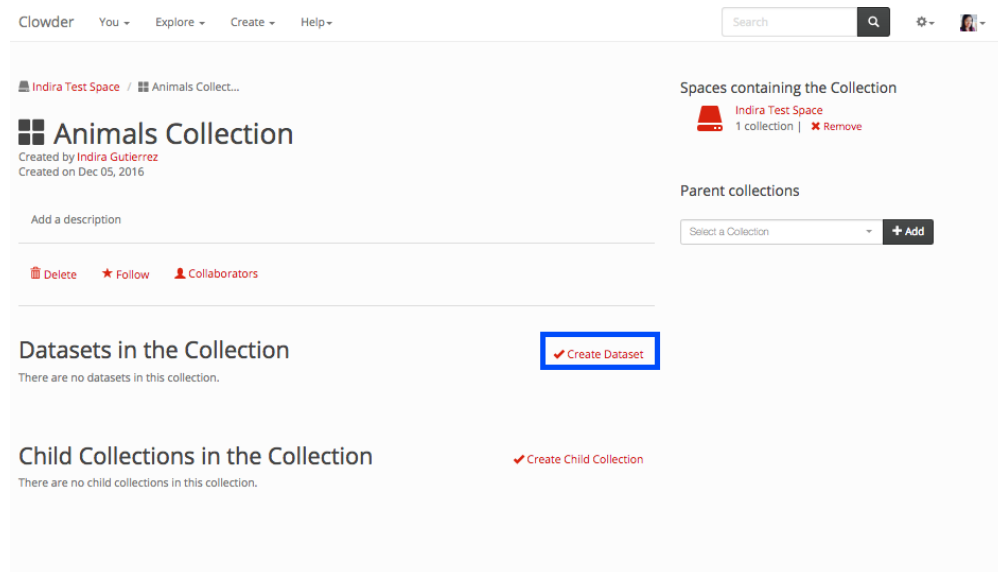

On click it will show an input field with the current name which you can update and then click on the save button. A similar behavior exists for the description field.

Adding Datasets to a Collection

If you want to add a new dataset to a collection, you can create it within the collection page by clicking the create dataset button highlighted in the image below.

You can create a dataset as indicated in the instructions on the dataset section.

If you want to add an existing dataset to a collection, you need to go into the datasets page. In the right navigation there is a section called 'Collections containing the dataset' where a list of the collections that a dataset is part of will show up, if it is part of a collection. At the end, there is a dropdown that will show the collections you can add the dataset to. You can select one and click on the Add button. The section where you can add a dataset to a collection is highlighted below in a blue box.
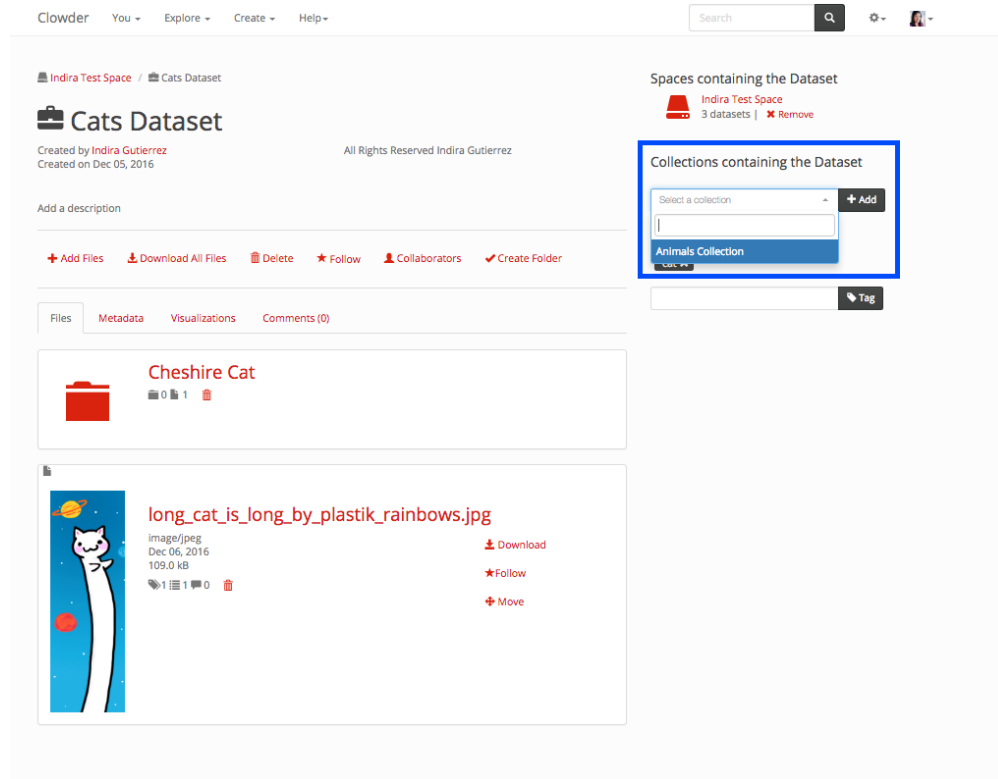
Removing a Dataset from a Collection

You can remove a dataset from a collection within the dataset page or within the collection page.

To remove a dataset from the dataset page, In the right navigation in the 'Collections containing the dataset' you can click on the remove button next to the collection you want to remove the dataset from. The remove link is highlighted in the next figure.
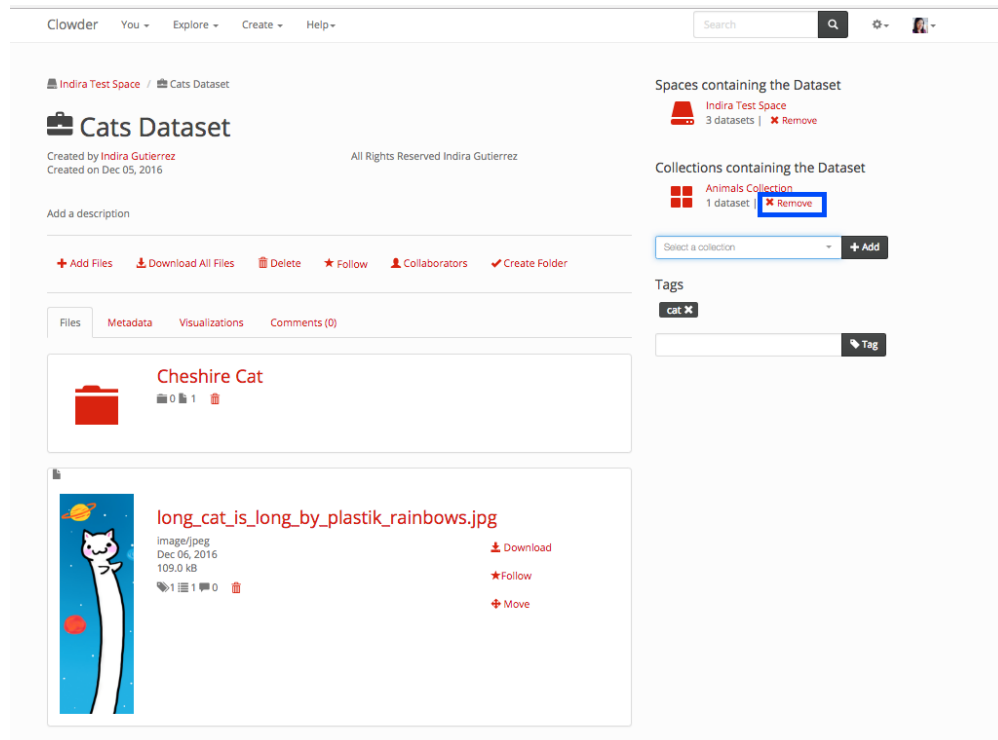
To remove a dataset from the collections page, in the list of dataset, you can click on the remove button in the dataset list item. The button is highlighted in the image below with a blue box.
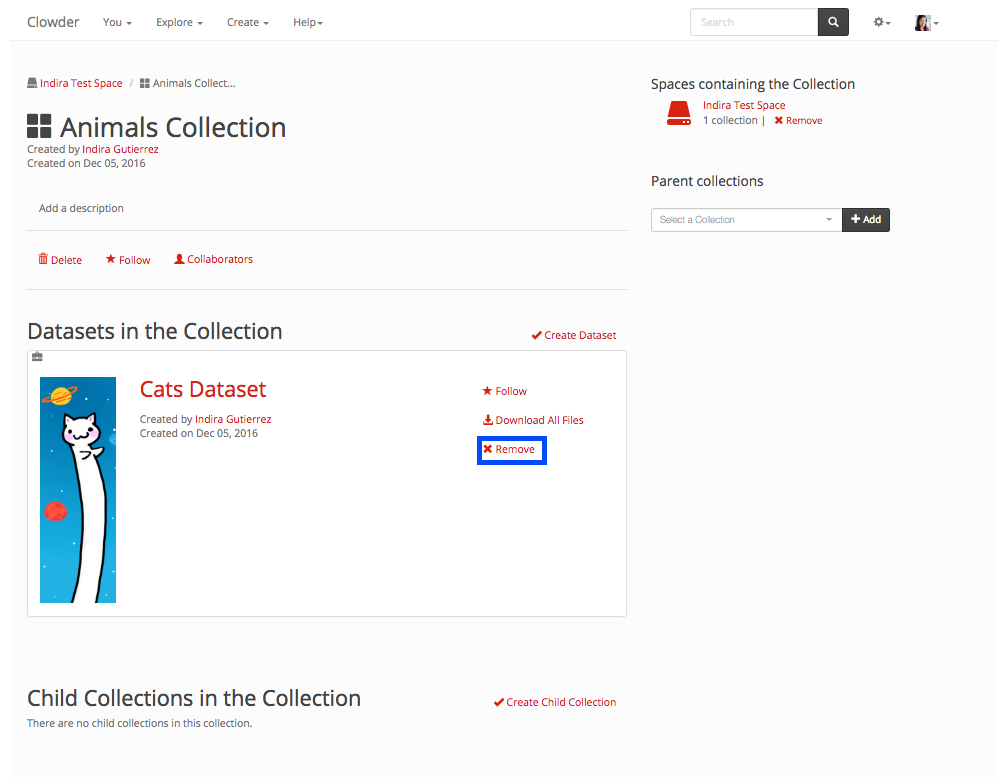
Creating Child Collections

Child collections are a way to organize collections in a hierarchical way. You can create new child collections within a collections page. Collections created this way inherit the spaces that the parent collection is currently on. Child collections do not show up on the listing of collections to avoid cluttering, but they do show up as collections that you created in the home page. You can not remove child collections from a space that has been inherited from a parent collec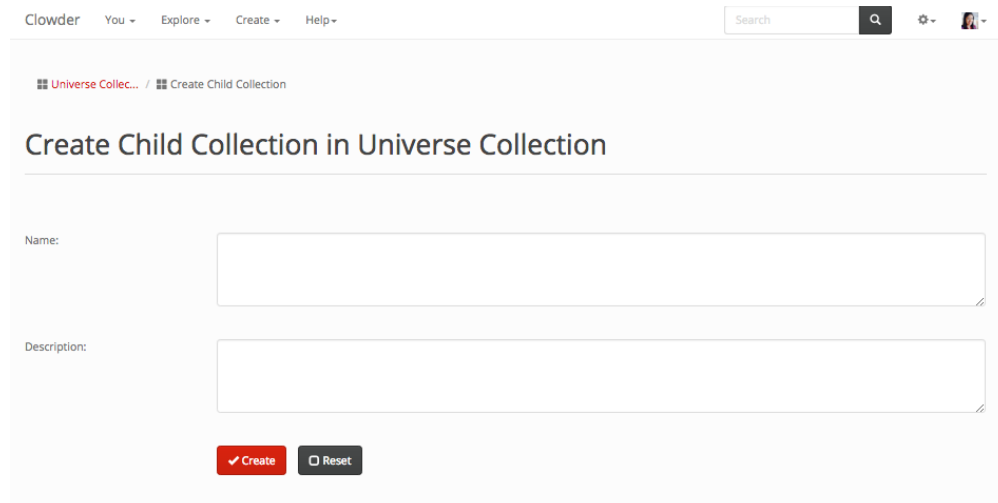tion. You can also add existing collections to a parent collection. If the child collection was on a space that the parent collection was not in, you will be able to remove the child collection from that space.

To create a child collection within a collection page. Click on the 'Create Child Collection' button in the Child Collections in the Collection section. The button is highlighted in a blue box in the next image.
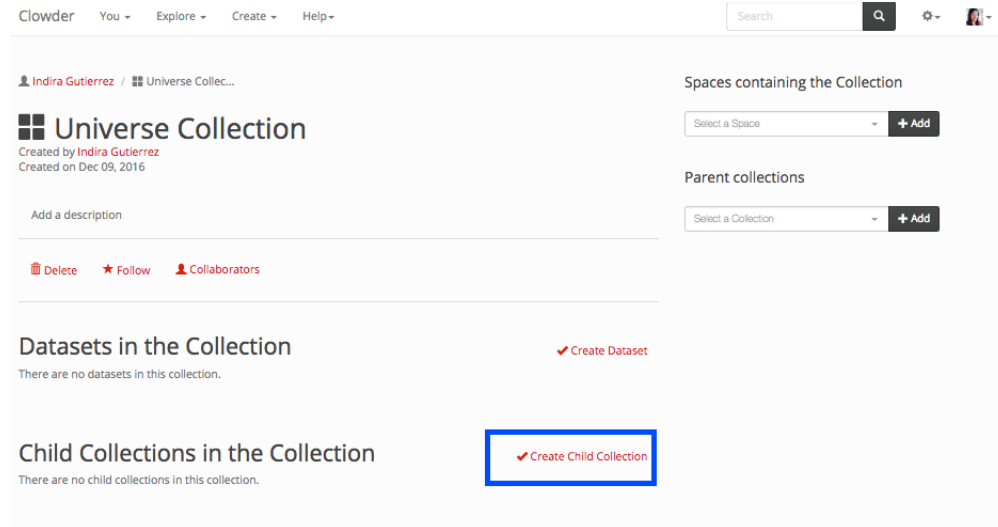
The Create Child Collection page looks as below. It is a little different than creating a collection page, because it cannot be added to a space directly. It will inherit the spaces from the parent collection. After adding a name and optional description you can click on the create button in the bottom.

To add an existing collection to a parent collection. You can click in the dropdown in the 'Parent Collections' area in the right navigation of the collection page. After selecting a collection from the dropdown, click on the Add button next to it. The Parent Collections area is highlighted in a blue box in the next image.
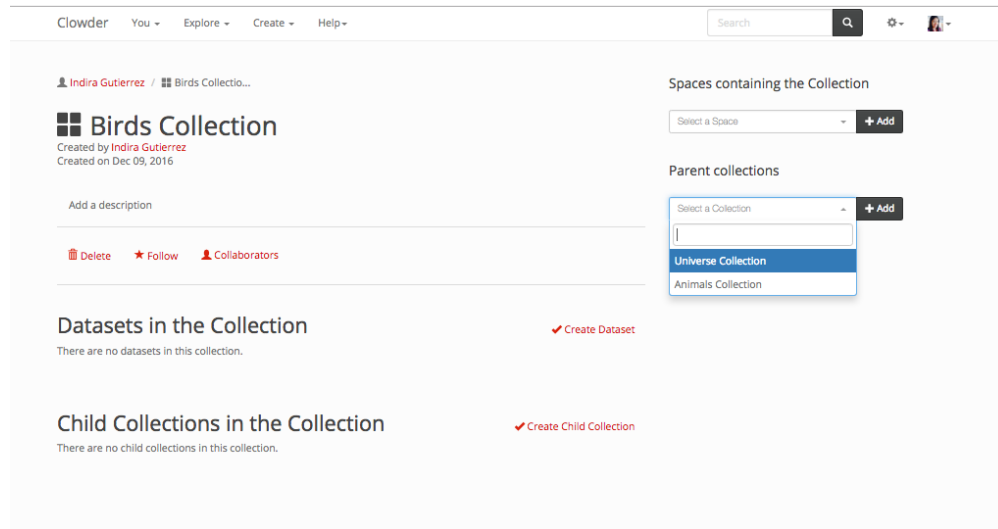
Removing a Child Collection

A child collection can only be removed from within the parents collection page. In the listing of Child Collections within the collection page, click on the remove button on the tile for the corresponding collection. In the next image the remove button for the child collection is highlighted.
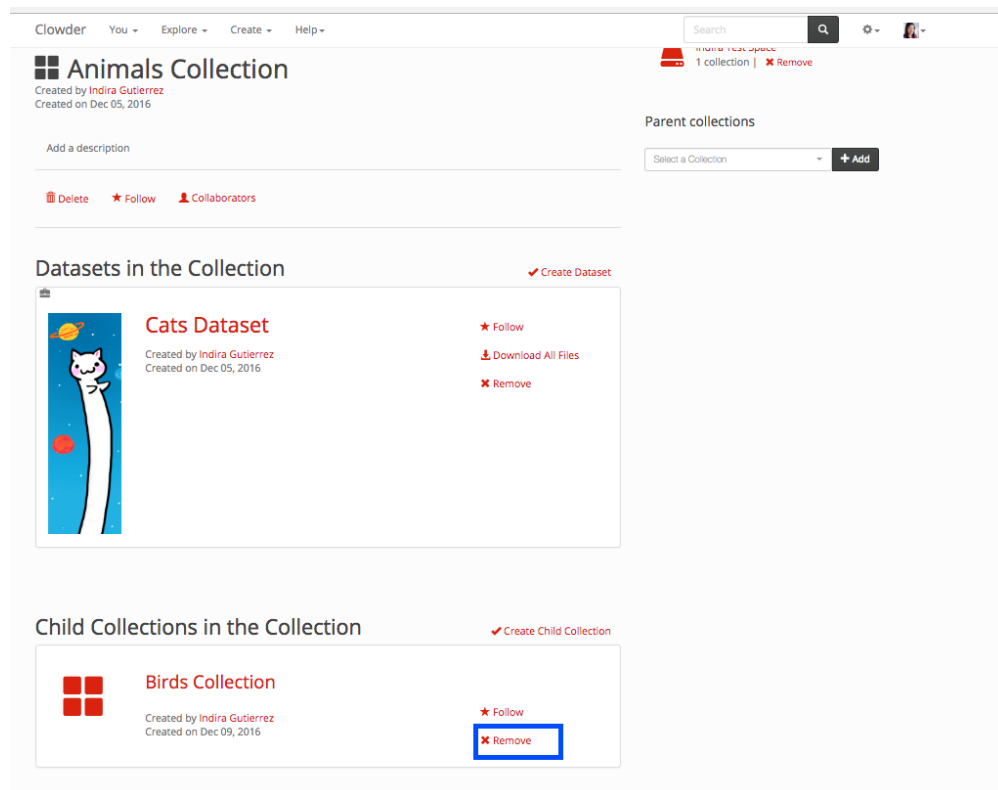
Adding to a Space

You can add an existing collection to a space by going to the collection page and clicking on the dropdown in the 'Spaces containing the Collection' section, selecting the space you want to add the collection to and then clicking on the Add button next to it.

The section where you can add a collection to a space is highlighted by a blue box in the next image.
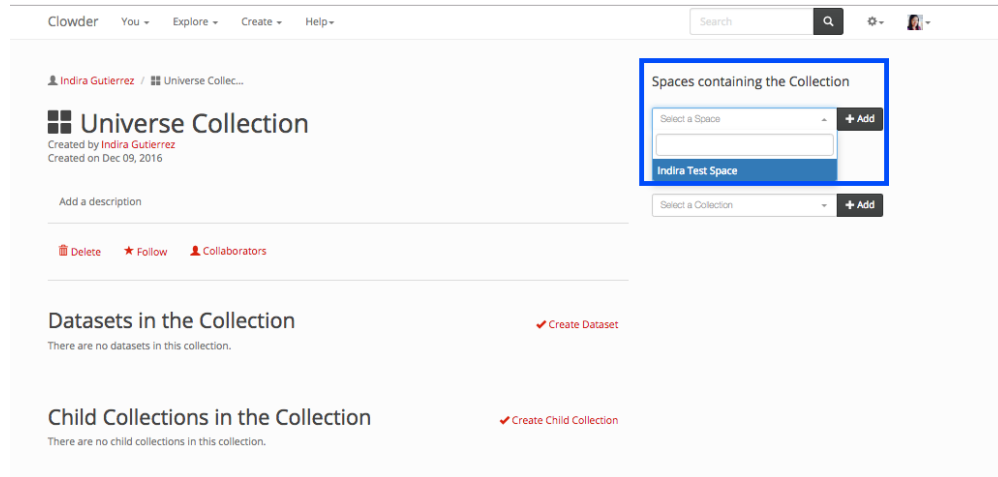
Removing from a Space

A collection can be removed from a space within the collection page or within the space page. To remove the collection within the collection page, click the 'Remove' button next to the space you want to remove it from. This is highlighted in the next image with a blue box.
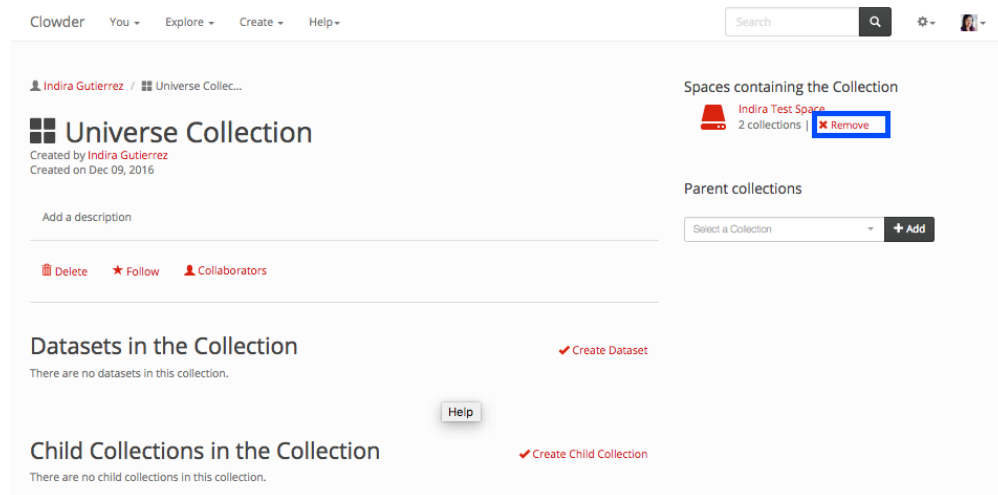
In a space page, go to the collection tile that you want to remove and click on the x button within the tile. The x button's for collections are highlighted in the image below with a blue box.
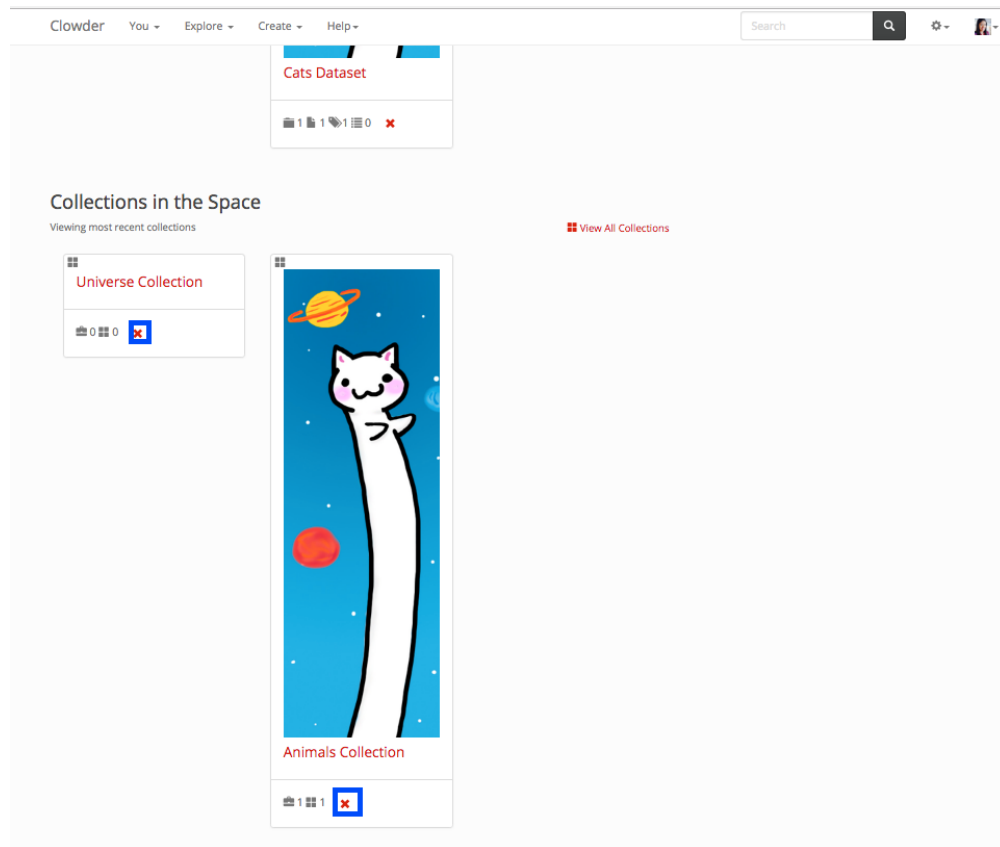
Deleting a Collection

If you want to completely delete a collection from Clowder, you can do so within the collection itself, on the collection list, or if you created the collection from the home page.

Within the collection, page click on the Delete button displayed below in a blue box.
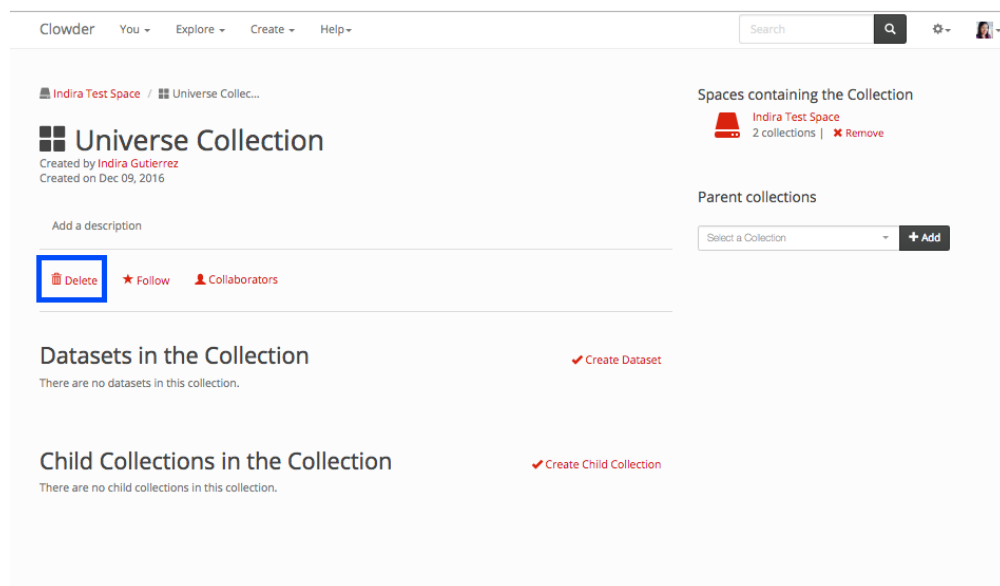
To delete a collection from the collection list page, click on the trash button within the tiles. They are highlighted below in blue boxes.
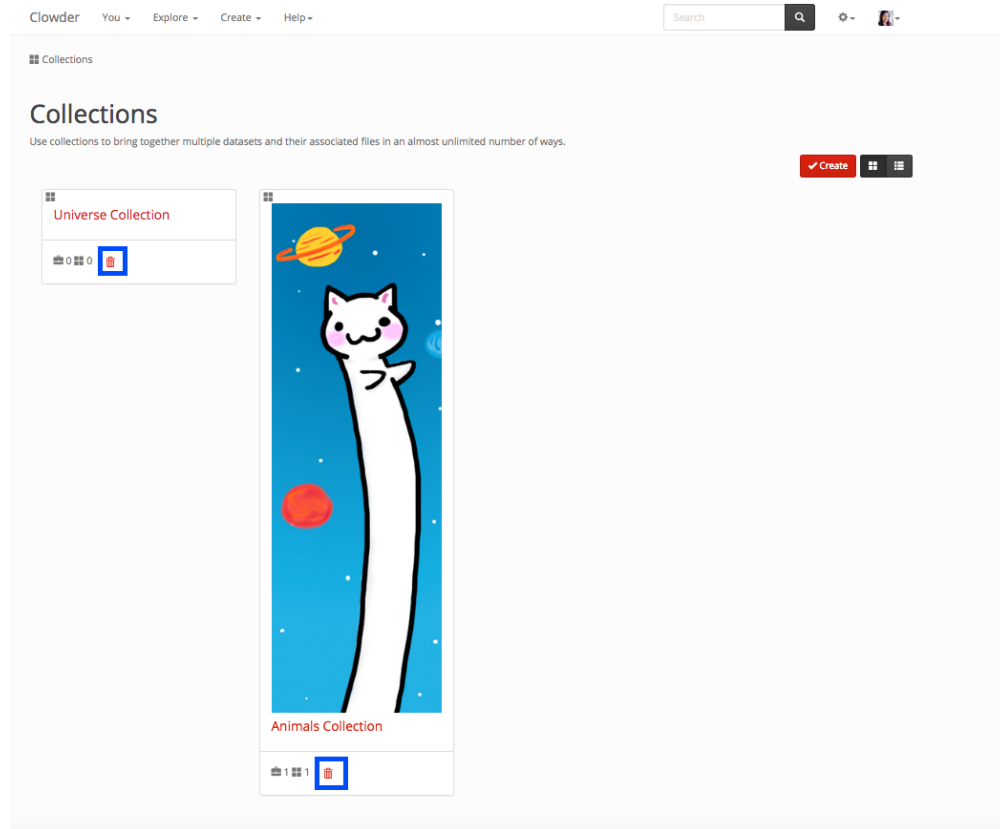
If you created a collection, you can delete them from your home page. Go into the 'My Collections' tab displayed below with a blue surrounding box. And then click on the trash can icon in the collection you want to delete, highlighted in green in the image below.
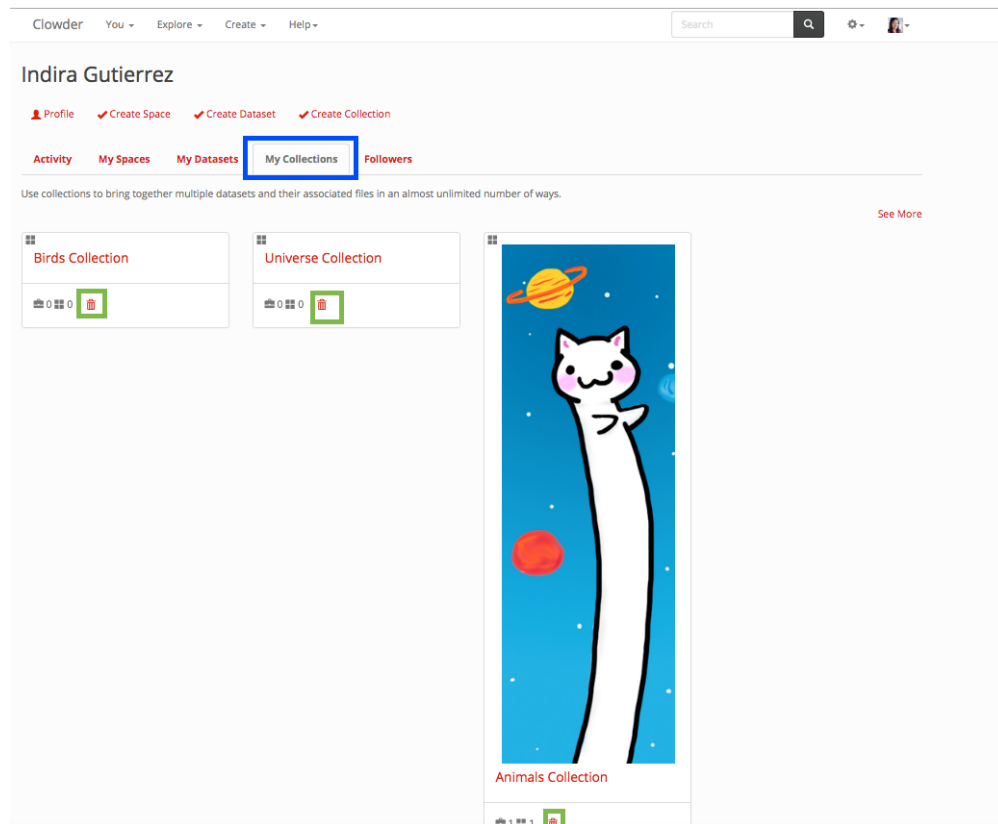
1.4.5 Spaces

A space is a group of collections, data sets, and files with defined user access rights. Spaces are used to share data within datasets and collections with other users. There are different permissions assigned to each role. The 3 most common roles are Admins, Editors, and Viewers. In summary, a viewer can only see the datasets, files and collections within a space. An editor, besides the view privileges can add new datasets and collections to the space, as well as remove them, it can also edit the datasets and collections within the space. An admin can do what an editor does and also edit the space itself, invite and remove people from the space, edit the extractor available. Note: Roles are customizable there is a section below about permissions and roles.

Creating a Space

You can create a space from your home page, by clicking on the button in the blue box below
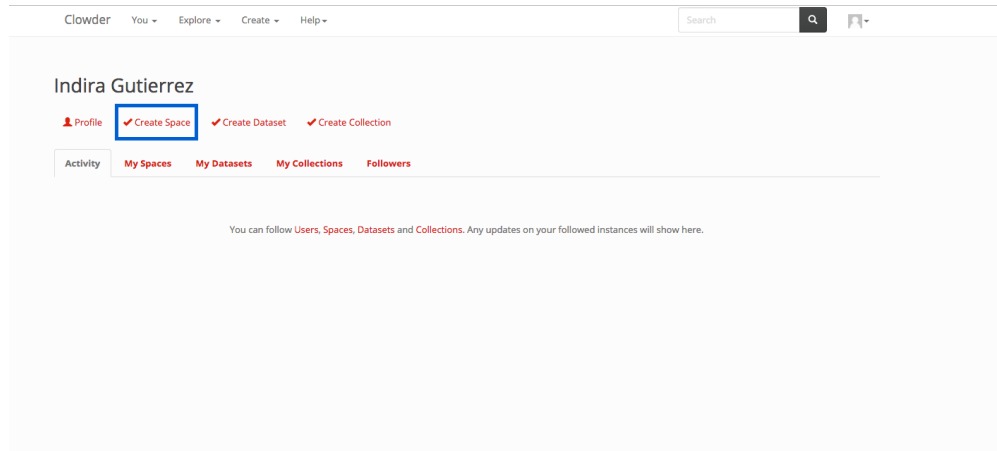
Or using the create dialog in the top navigation. Click on the Create dropdown in the top navigation (in blue on the image below) And then select Spaces (in green in the image below)
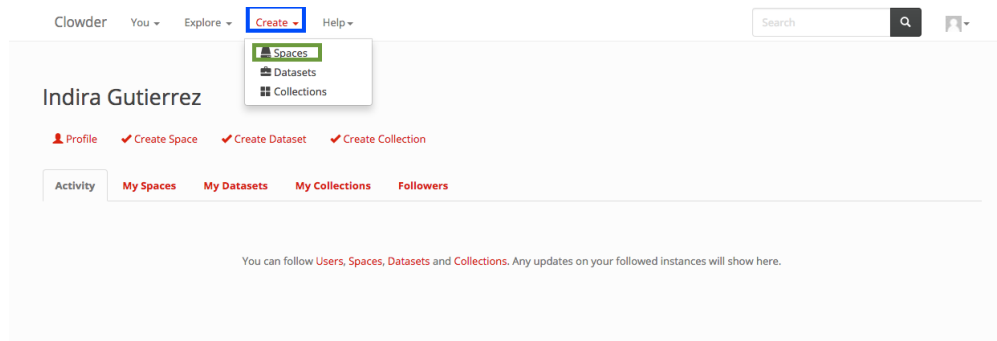
Or within the space list page. Click on the Create button in the top right of the page. (The list spaces page is accessed by clicking on Explore > Spaces)
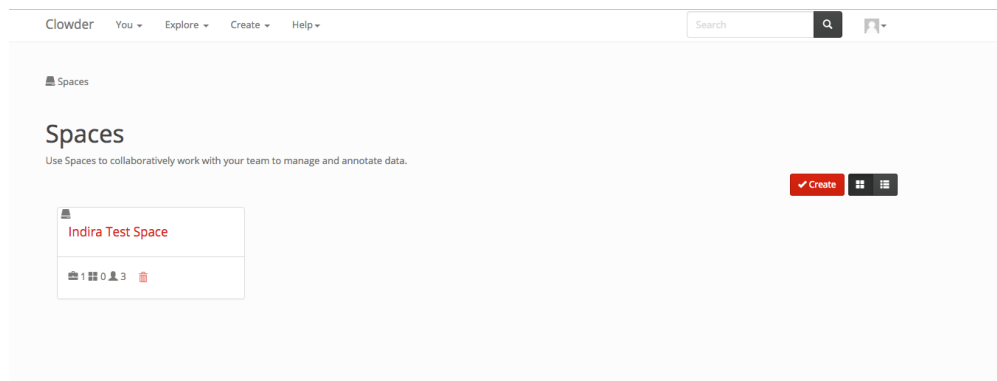
After starting the create process by either of the 3 methods above you can start creating your space by filling up the information in the next figure (The only required field is the name).

You click on the create button at the end of the page and are then redirected to the space page.

Editing a Space

You can edit the name, description, external links, logo and banner for your space at any time. To do so, click on the Edit Space button highlighted in blue in the figure
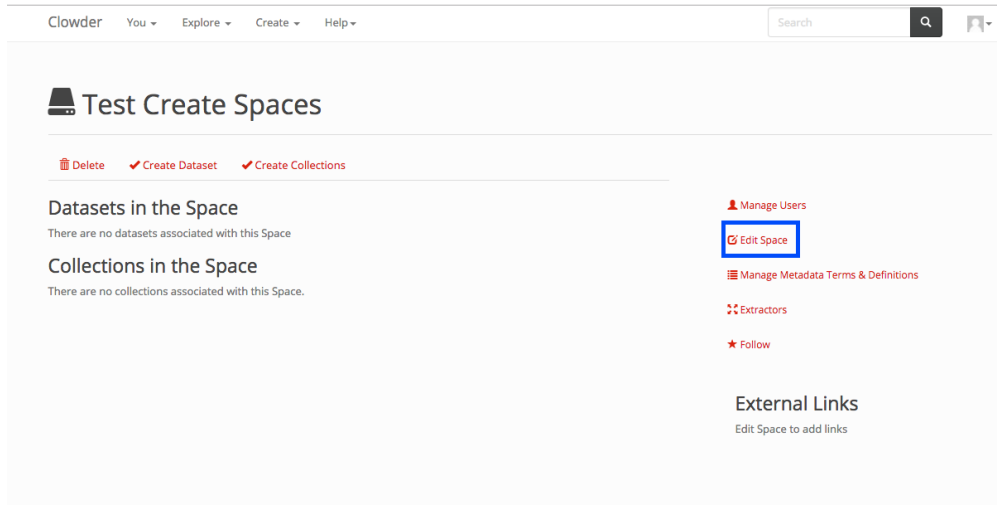
Then you can edit the values and click on the update button in the bottom.
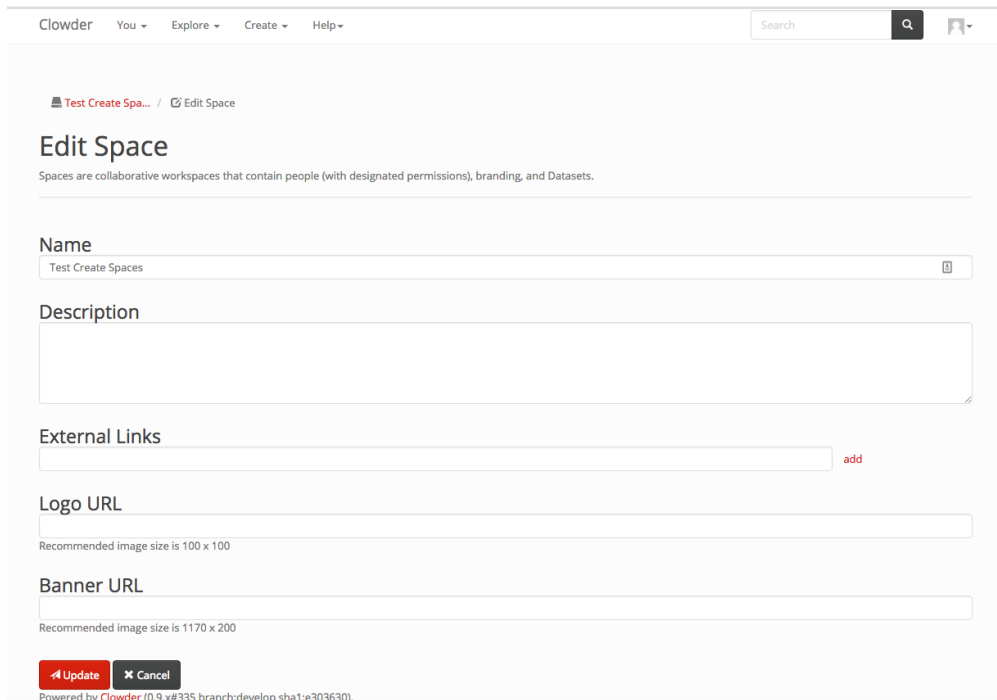
Inviting Users and Adding users to a Space

When you want to invite users to collaborate in your space, you can invite them by email if they don't have an account on Clowder, and if they have an account on Clowder you can add them to your space. In order to do so, in the space page click on the Manage Users button in the right column, as marked by the blue box in the next figure.
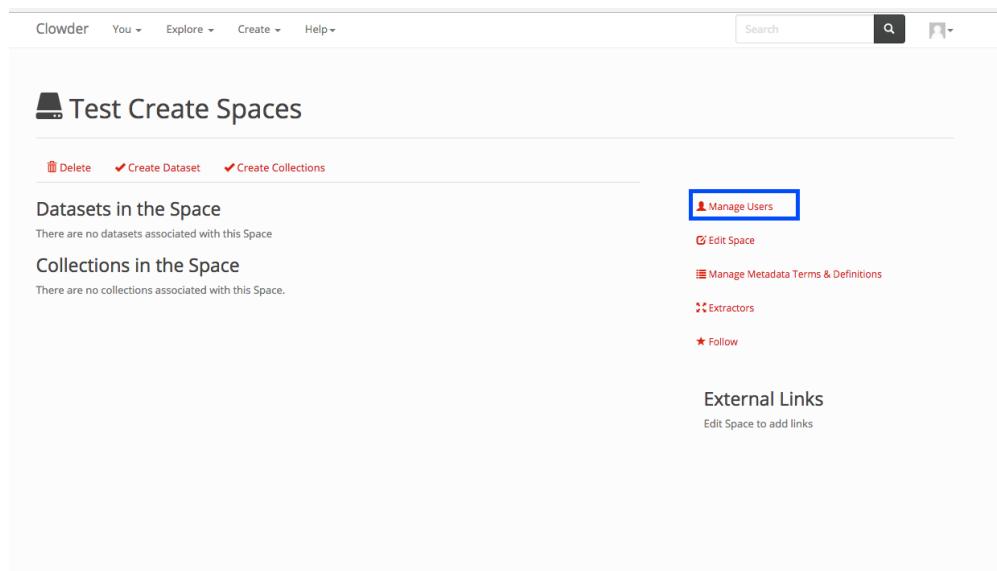
After clicking on the link above, the screen below shows up. You can click on either of the 3 roles select fields, and when you do so, the list of all the members of clowder not in the space show up. You can select them. And after selecting all the people you want in the different roles you click on the submit button at the end of the page. In this page, you can also remove current members, there is an x next to each current member of the space. When you click on the x, the member is removed from the space (no need to click on submit).
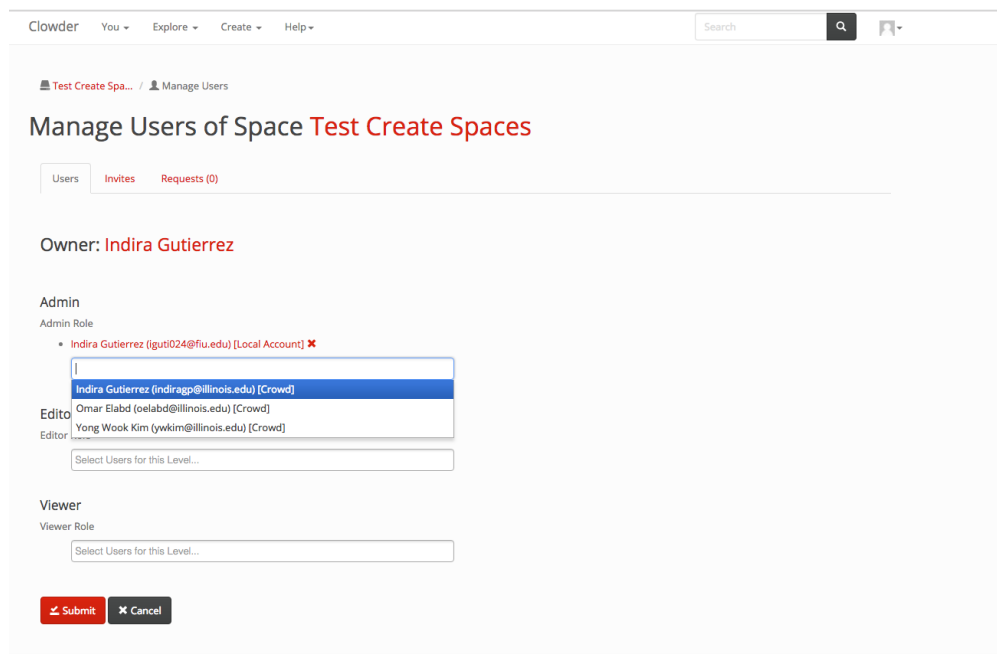
To invite people by email, click on the Invite shown in a blue box in the image below.
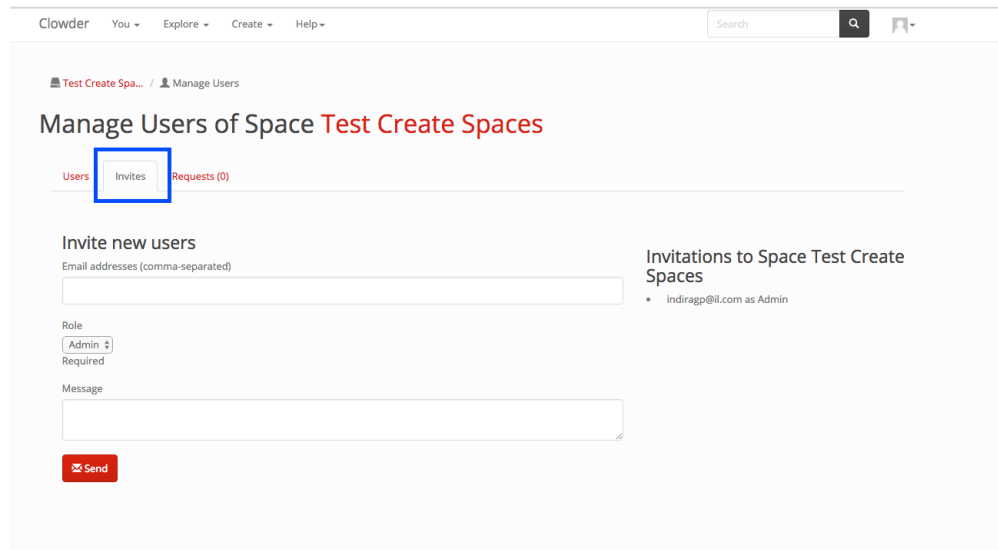
You can fill up the emails for the users you want to invite, select the role and an optional message. The people you invite will get a link to register to clowder and will get added to your space once they join clowder. The current invites you have out for people show up in the right with the roles they were invited as.

List of All Spaces

To access a list of all the spaces, you click on the explore button in the top navigation (in blue in the next screenshot). And then click on the spaces button (in green in the next screenshot). The list of all available spaces in the space will be displayed.
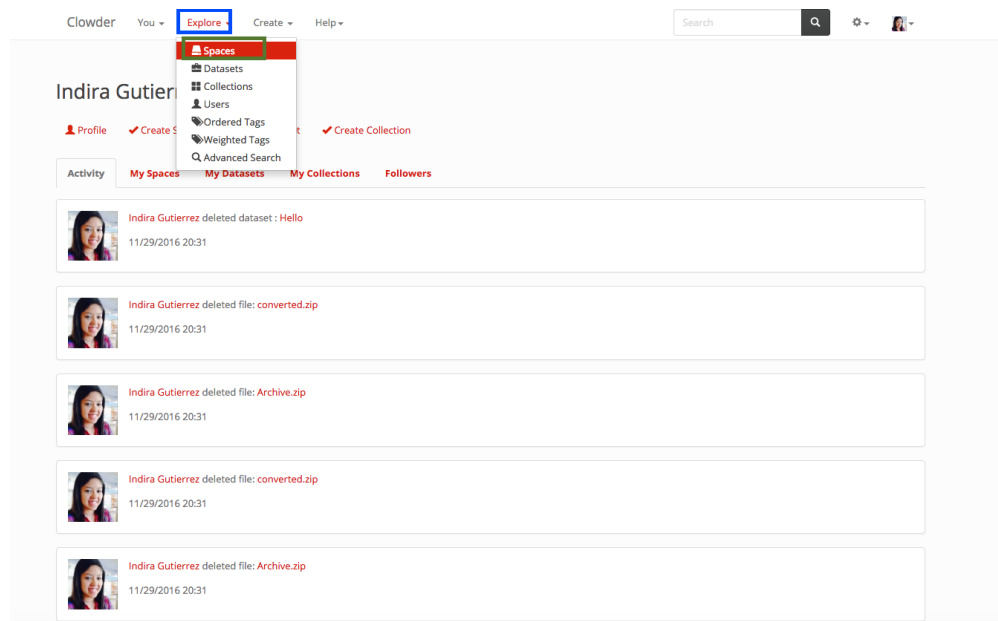
Requesting and Granting Access to a Space

If there is a space that you would like to participate in, but you are not a part of, you can request access to it. In the space page you click on the button pointed by the green arrows in the image below.
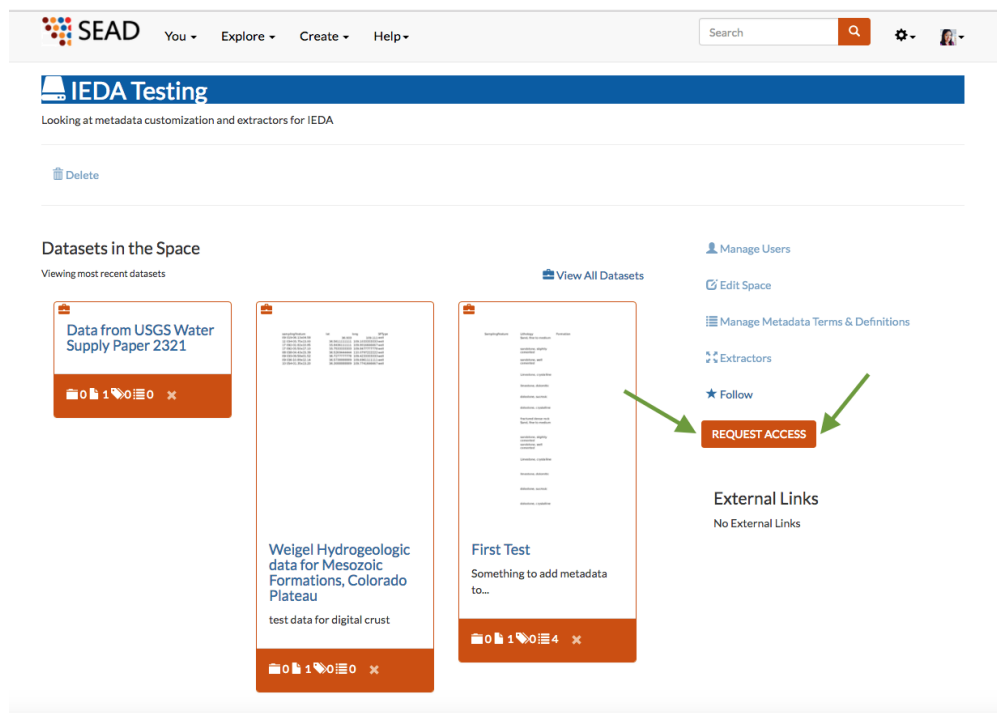
When you submit your request, the admins of the space will get an email and can then accept you to the space.

To accept people that have requested access to one of your spaces. You first go to the space and then click on the Manage Users (as when inviting people above)

Deleting a Space

A space can be deleted within the list of spaces or inside the space itself. If you are also the creator of the space you can delete it from your home page. To delete a space from the list of the space. Go to the list of spaces, as indicated above, click on the explore dropdown in the top navigation, then select spaces. If you have the right permissions to delete the space the delete button will be enabled. In the screenshot below the delete button is shown in a blue box for the 2nd space.
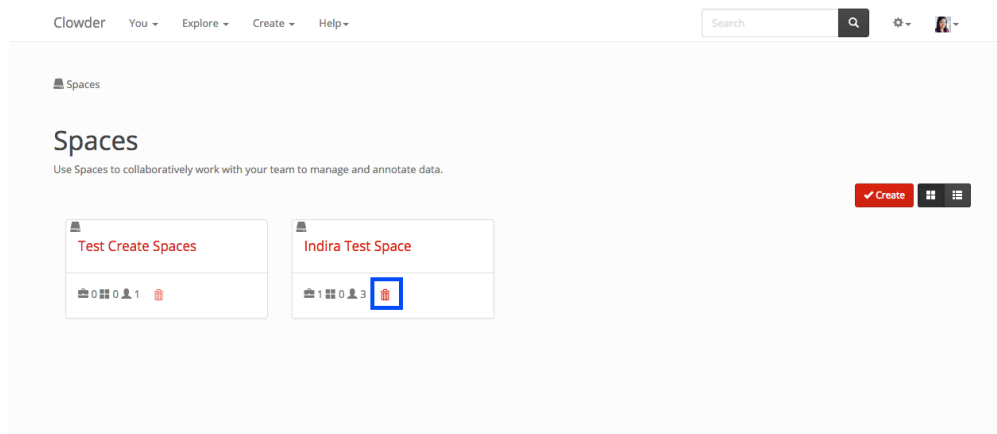
You can click on a space within the list of spaces, or the home page and when you are on the space itself, you can click on the delete button indicated by a blue box surrounding it in the next picture
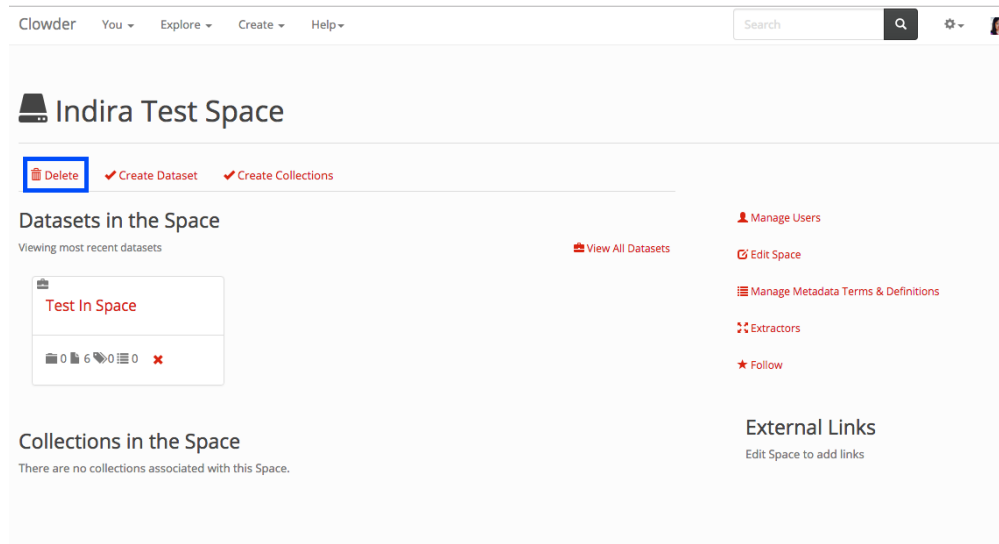
If you are the creator of the space you want to delete, you can go to the home page, click on the my spaces tab and delete like in the first scenario. The spaces tab is highlighted in blue in the next picture and the delete button in green.
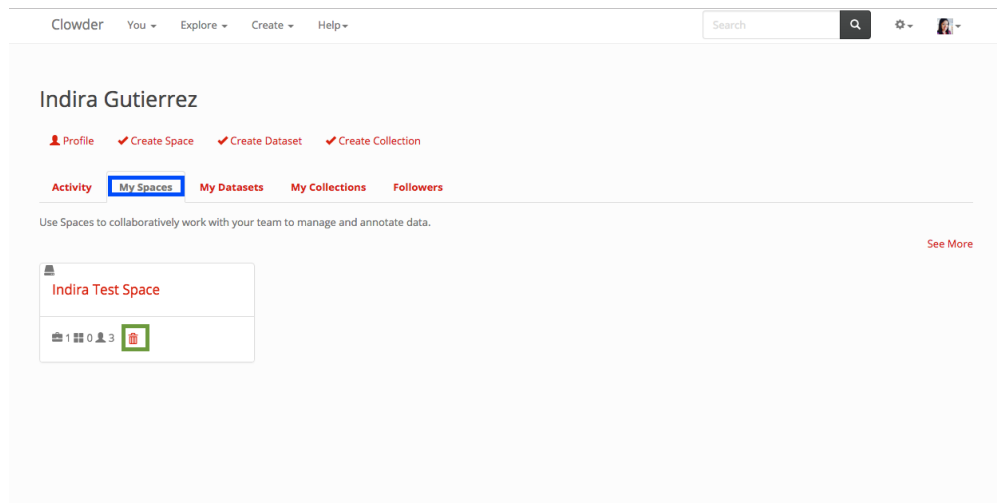
1.4.6 Search

In Clowder, you can search datasets, collections and files by name and description. To do so, just input the string you want to look for in the search box on the top right. As indicated in the image below with a blue box and click on the search button next to it or click enter. The image below also shows a sample result of a search.

Advanced Search

There is an advanced search which allows users to search by metadata. This is available by clicking Explore in the top navigation and then clicking Advanced Search

This page allows users to search by metadata, it lists user defined metadata fields as indicated by the open dropdown

in the next figure. It also allows users to search on extractor metadata, but the user need to start typing the name of the extractor metadata field in order to select it. The results are files and datasets that match the criteria. The search can have multiple terms and the results could match all the selected terms or any of the selected terms.
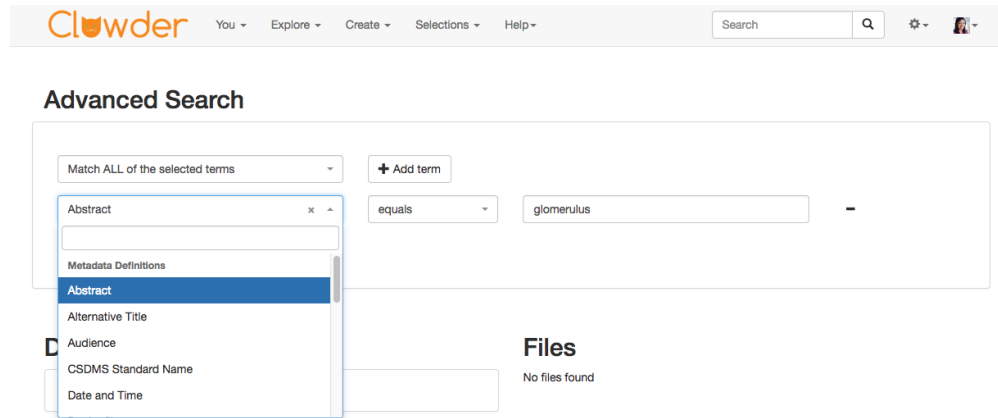
1.4.7 Following

You can follow spaces, collections, datasets, files and users.

You can follow a space within a space page. The follow button is highlighted in a blue box below. When you are already following the item the location of the unfollow button is the same as the follow button.
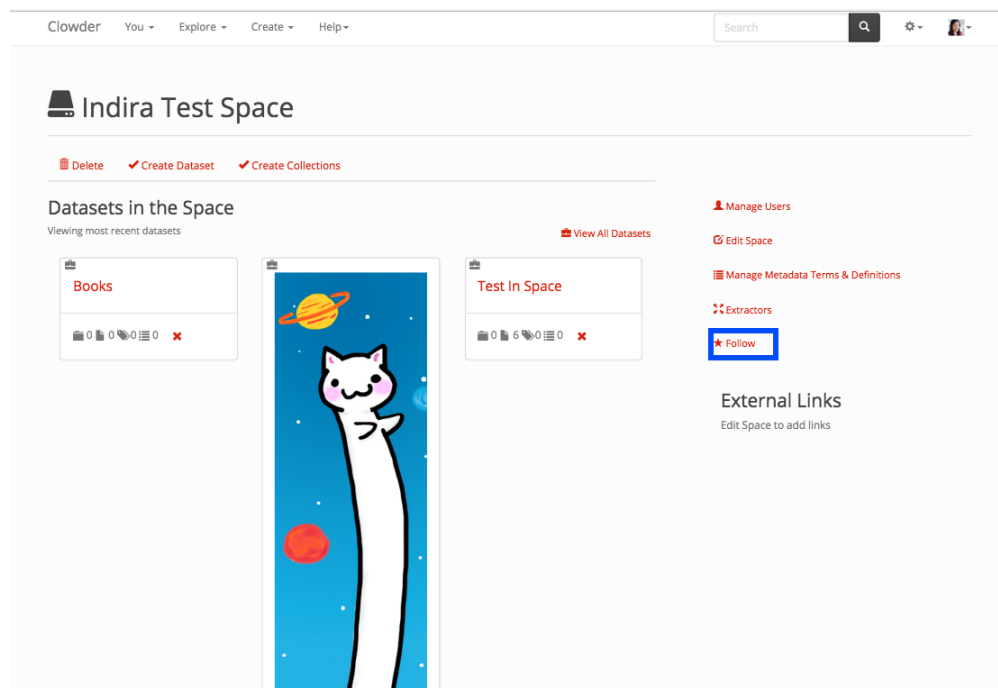
To follow a dataset go within a dataset page and click on the follow button displayed in blue in the next image.
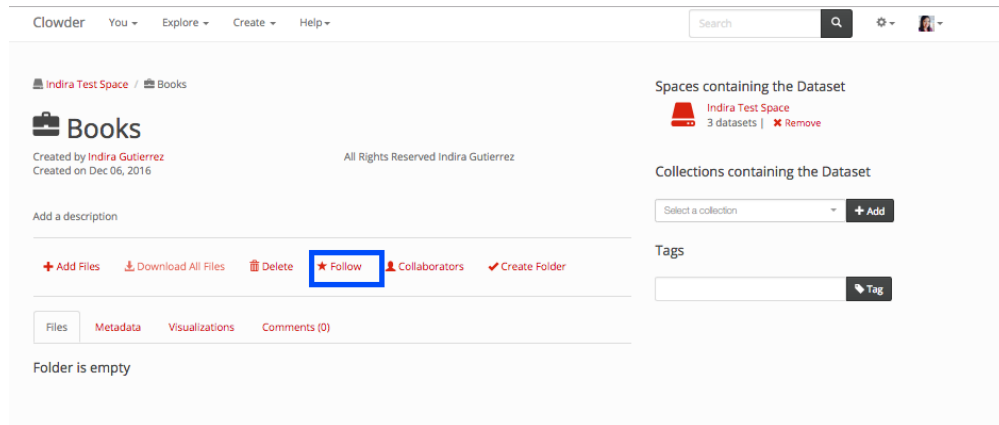
To follow a collection within a collection page click on the follow button displayed in blue in the next image.
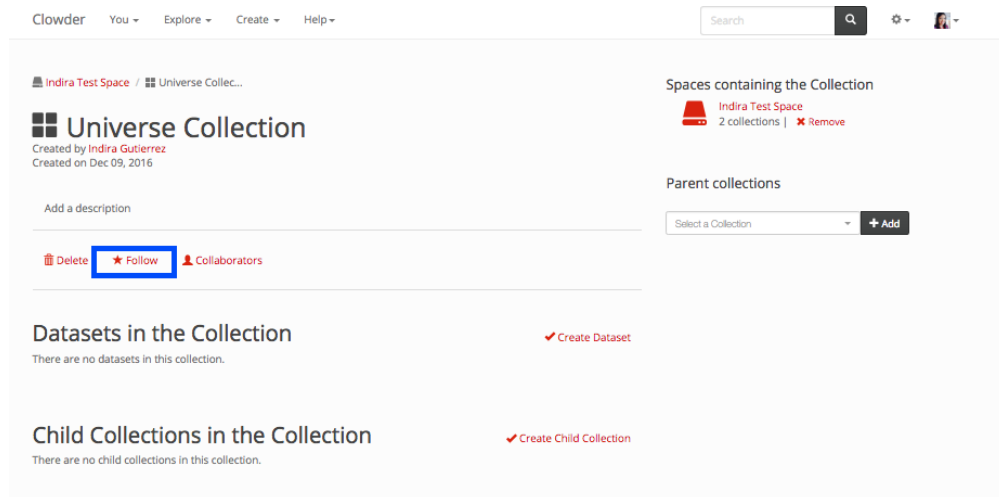
To follow a file within the file page click on the follow button displayed in blue in the next image.
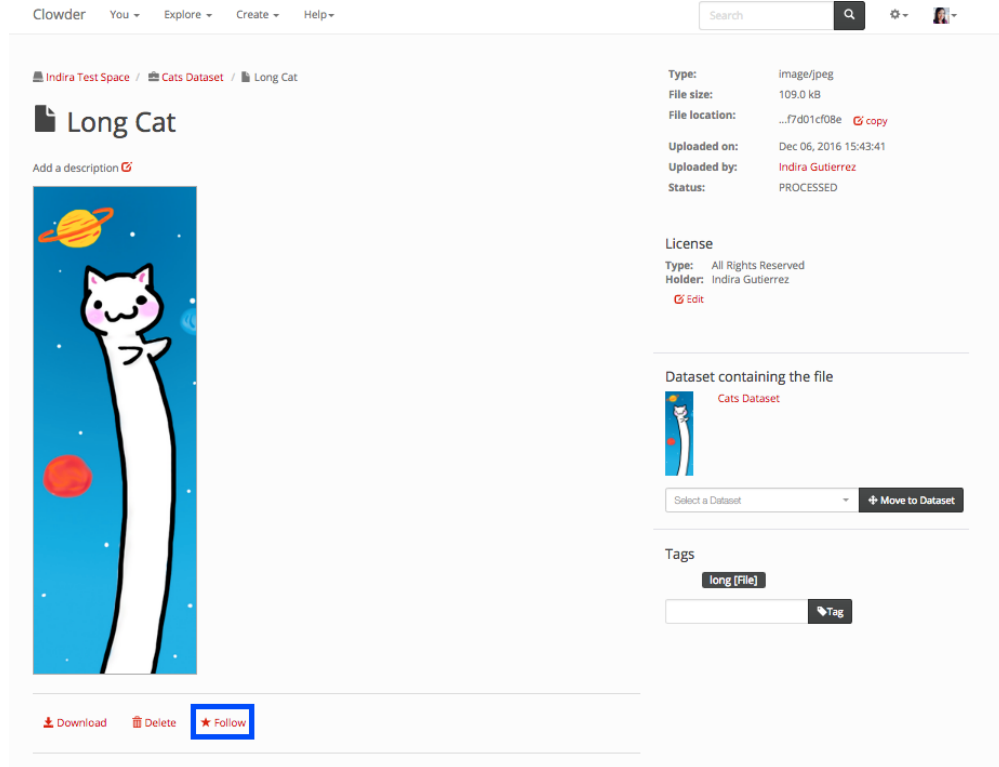
To follow a user, you can go into the listing page of users in by clicking on Explore in the top navigation as indicated in blue in the next image and then users as indicated by the green box.
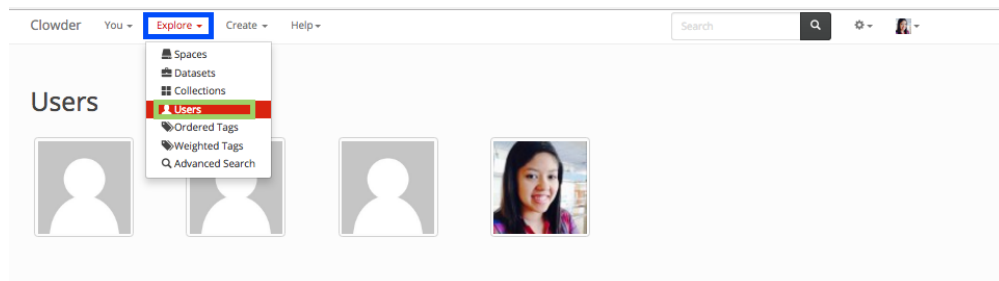
Then on hover in a user, you can click the follow button as indicated by the blue box.
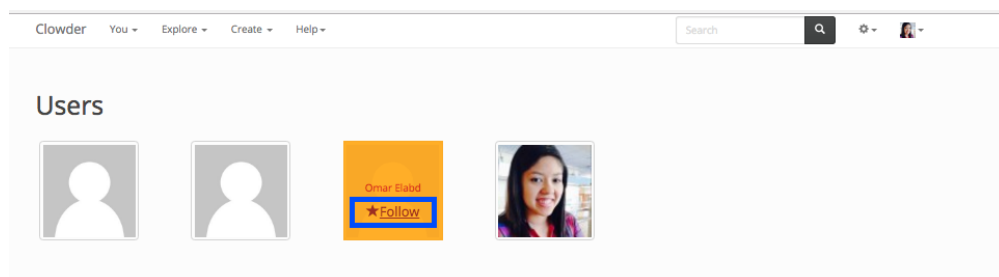
1.5 Installing Clowder

1.5.1 What type of user are you?

For **most users of Clowder**: *Get started here*

For **developers of Clowder itself**: *Dev quickstart here*

- Build Clowder from source via IntelliJ's Play-2 run configuration.

- Run the required services via Docker.

For **production instances of Clowder**, a Kubernetes deployment is recommended and manual installations are being phased out.

- See Clowder's Kubernetes install instructions here.

- Otherwise, manually install Clowder, and each of it's required services (at a minimum: MongoDB, Elastic-Search, RabbitMQ). See *requirements below* for details.

1.5.2 Users of Clowder: Getting Started via Docker

1. Install Docker Desktop (if you haven't already)

2. Clone or download Clowder on Github (use the default `develop` branch)

```
git clone https://github.com/clowder-framework/clowder.git
```

3. Navigate to Clowder's root directory in your bash command line (`cd clowder`)

4. Start Clowder using the Docker Compose configuration, via your command line

```
docker-compose up -d
```

5. If you experience any issue with file uploads and see the below error message in the console:

```
[ERROR  ] - application - Could not create folder on disk /home/clowder/data/uploads/
→xx/xx/xx
[ERROR  ] - application - Could not save bytes, deleting file xxx
```

you can try this command:

```
docker-compose exec -u 0 clowder chmod 777 /home/clowder/data
```

6. Open your web browser to `localhost:8000`

- If you see Error 404, allow a minute for it to appear.

- Note: use port 8000 for docker run. However, port 9000 for manual builds.

If you experience *any* trouble, come ask us on Slack here!

Helpful docker commands

- `docker-compose up -d` - start up all required services

- `docker-compose down` - stop all docker containers

- `docker-compose logs -f` - see the logs

- `docker ps` - check how many services are running

- `docker info` - details about your docker version

  - After starting Docker, check that your services are running via the Docker Desktop GUI, or run `docker ps` and check that 3 containers are running.

  - The "image" column should show `rabbitmq`, `elasticsearch` and `mongo`.

Clowder started! Now *create a new user*

Sign up for a Clowder login account

After installing Clowder, you still need to sign up for a user account.

Run this in your terminal to create a new account:

```
docker run --rm -ti --network clowder_clowder -e \
    FIRST_NAME=Admin -e LAST_NAME=User \
    -e EMAIL_ADDRESS=admin@example.com -e PASSWORD=catsarecute \
    -e ADMIN=true clowder/mongo-init
```

Optionally, edit these properties to your liking:

- FIRST_NAME

- LAST_NAME

- EMAIL_ADDRESS

- PASSWORD

- ADMIN (only set this if you want the user to have superadmin rights, make sure at lease one user has this).

Configuration complete! Now you can login to Clowder via `localhost:9000` in your browser.

> **Warning:** If you renamed the base clowder folder to something else, like *kitten*, then the `--netowrk` parameter must be changed to `--network kitten_clowder`.

All done! You should be able to login to your new account, create new Spaces & Datasets and upload many different types of data.

Note: Before you go, check out useful information like the Clowder 'All Paws' YouTube playist.

- Check out the Overview of Clowder

- Check out How to use Clowder

- Check out How to Create a New Extractor

- and 28 total videos covering specific Clowder topics and uses!

Try the *default extractors* for simple quality of life improvements in Clowder.

```
$ docker-compose -f docker-compose.yml -f docker-compose.override.yml -f docker-
↪compose.extractors.yml up -d
```

1.5.3 Clowder Developers: Getting Started

For **Clowder developers**, a hybrid is recommended:

- Part 1: Run the required services via **Docker**, and expose each of their ports to Clowder.
- Part 2: Run the Clowder instance manually via **IntelliJ Ultimate**'s Play-2 run configuration.

Part 1: Setup Docker

1. Install Docker (if you haven't already)
2. Clone Clowder's `develop` branch (the default)

```
git clone https://github.com/clowder-framework/clowder.git
```

Apple Silicon M1 users, additional instructions here

Clowder works well on Apple Silicon, with only one minor caveat. No changes are necessary, but these optimizations are handy.

Elasticsearch does not work and so the search bar in the top right of the web interface will not work or be visible. Clowder depends on an older version of Elasticsearch before it added Apple Silicon support, and Docker's QEMU emulation of x64 happens to fail causing the container to infinitely crash and restart.

To prevent this container from constantly crashing and restarting, please comment it out of the Docker definition in `docker-compose.yml`.

```
# COMMENT THIS OUT in docker-compose.yml:

# search index (optional, needed for search and sorting future)
elasticsearch:
  image: clowder/elasticsearch:${CLOWDER_VERSION:-latest}
  command: elasticsearch -Des.cluster.name="clowder"
  networks:
    - clowder
  restart: unless-stopped
  environment:
    - cluster.name=clowder
  volumes:
    - elasticsearch:/usr/share/elasticsearch/data
```

Additionally, you may have to install Scala and SBT on your Mac.

```
brew install scala sbt
```

Finally, there is *no need* to specify a 'default Docker platform, and could hurt performance. (i.e. `<DO NOT> export DOCKER_DEFAULT_PLATFORM=linux/amd64`.') Only the necessary Docker containers will automatically emulate x64, and the rest will run natively on Apple Silicon.

Expose Docker services' ports to Clowder

In order for Clowder to access the required services (at a minimum: MongoDB, ElasticSearch, RabbitMQ. See Requirements for details.), we must tell Clowder which ports the services are using.

1. Create an override file, where we will store the port information

```
# navigate to Clowder base directory
cd clowder
# create new file docker-compose.override.yml
touch docker-compose.override.yml
```

2. Copy and paste the lines below into that file we just created `docker-compose.override.yml`

Default

Listing 1: docker-compose.override.yml

```yaml
# Enable Clowder to communicate with the necessary services (Mongo, RabbitMQ,
→ElsticSearch)
# Each service runs as a Docker container.

services:
mongo:
   ports:
      - 27017:27017
rabbitmq:
   ports:
      - 5672:5672
      - 15672:15672
elasticsearch:
   image: elasticsearch:2
   ports:
      - 9200:9200
      - 9300:9300
```

Apple Silicon M1

Listing 2: docker-compose.override.yml

```yaml
# Enable Clowder to communicate with the necessary services (Mongo, RabbitMQ,
→ElsticSearch)
# Each service runs as a Docker container.

services:
mongo:
   ports:
      - 27017:27017
rabbitmq:
   ports:
      - 5672:5672
      - 15672:15672
# Elasticsearch does NOT work with Apple Silicon M1. Do not include it here.
# That's okay, but as a result the search bar will not be visible.
```

(continues on next page)

<div align="right">(continued from previous page)</div>

```
# elasticsearch:
# image: elasticsearch:2
# ports:
#   - 9200:9200
#   - 9300:9300
```

3. Test that our services work! First start them:

```
docker-compose up -d
```

Note: By default, running `docker-compose up -d` uses the `docker-compose.yml` configuration **and will apply overrides found in** `docker-compose.override.yml`. Neither file need to be specified on the command line.

(Optional) Check that the Docker containers are running

You can see them in the Docker Desktop application, or in the web browser shown below.

`localhost:27017` - You should see: "It looks like you're trying to access MongoDB..." Success! `localhost:15672` - You should see: the RabbitMQ login screen (no need to login tho!). Success!

Now keep everything running, and next let's build Clowder from source

Part 2: Run Clowder via IntelliJ

1. Install IntelliJ **Ultimate Edition**.

   • This guide will assume developers use IntelliJ. Ultimate Edition is required for the Play2 configuration.

2. Open the base Clowder directory & install Scala plugin

   • This should prompt you to install the Scala plugin! Install it.

   • Or, manually install the Scala Plugin for IntelliJ `File –> Settings –> Plugins –> Download Scala`.

3. Install Java 8 (i.e. Java 1.8) on your computer. Clowder requires Java version 8 and is not compatible with other versions.

   • I find this easiest to do via IntelliJ's Plugin Manager. `File –> Project Structure –> SDKs –> +` icon `–> Download JDK`

   • Select `Version 1.8` (Clowder is only compatible with Java 8 (1.8), nothing higher) `–>` Vendor: `Eclipse Temurin (AdoptOpenJDK Hotspot) –> Download`.

   • Alternatively, download the JDK online at AdoptOpenJDK. Java 8, HotSpot.

   • Then point IntelliJ to the JDK folder under `Project Structure –> SDKs` and specify the root folder of the JDK you just downloaded.

4. Add a new Run Configuration

   • In the top right, click the dropdown and click "Edit Configurations..."

5. Create a new `Play 2 App` configuration
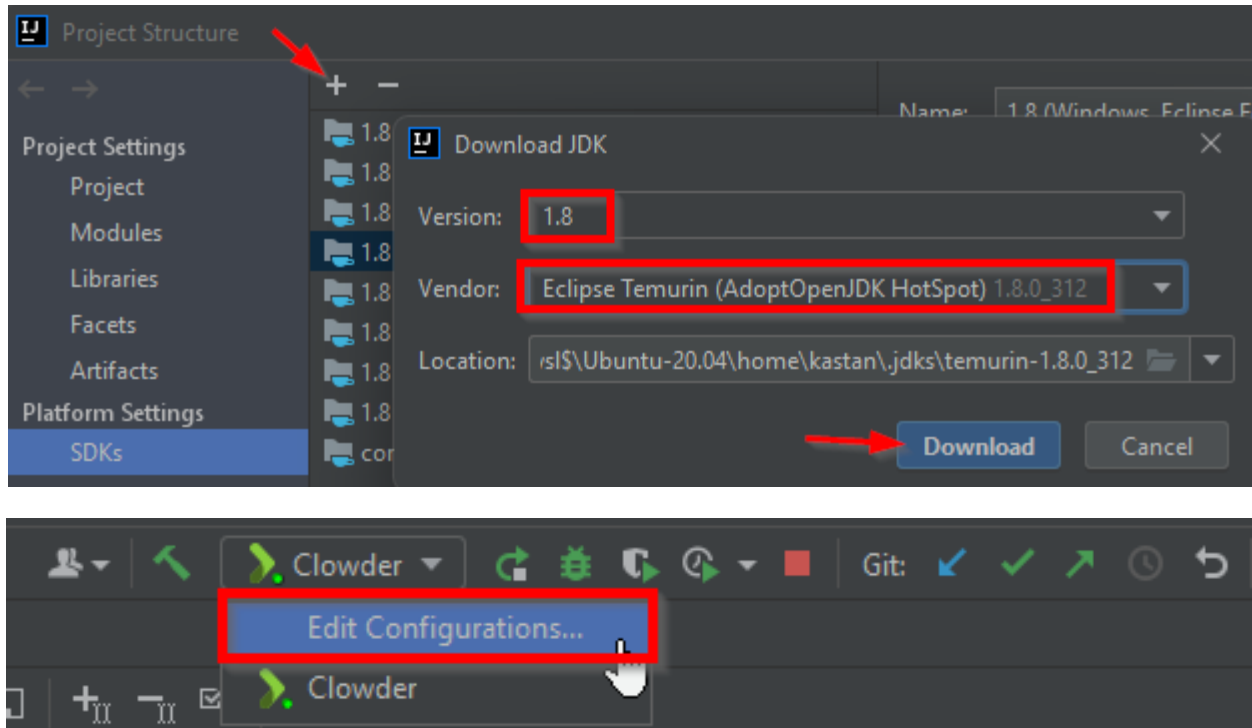
Note: If you **don't see Play 2 App in the list**, ensure you have the Scala plugin installed. If Play2 still isn't there, you may need to use IntelliJ *Ultimate* version (instead of Community). I experienced this bug, feel free to ask in the Clowder Slack here.

---

6. The default run configuration should be okay, see image below.

Note: Later, **if Clowder feels slow** (multiple seconds per page load) then you will need to add JNotify to your JVM Options on this page. *See the instructions at bottom of this page*.

Now start Clowder: In IntelliJ, click the green play button (top right) to build Clowder from source! Give it a minute to finish. Access Clowder via `localhost:9000` in the browser.

Also note, a handy debugging mode is enabled by default. You can run the debug mode by clicking the green "bug" button right beside the play button.

Creating a local Clowder account

After installing Clowder, you still need to sign up for a user account.

Run this in your terminal to create a new account:

```
docker run --rm -ti --network clowder_clowder -e \
  FIRST_NAME=Admin -e LAST_NAME=User \
  -e EMAIL_ADDRESS=admin@example.com -e PASSWORD=catsarecute \
  -e ADMIN=true clowder/mongo-init
```

Optionally, edit these properties to your liking:
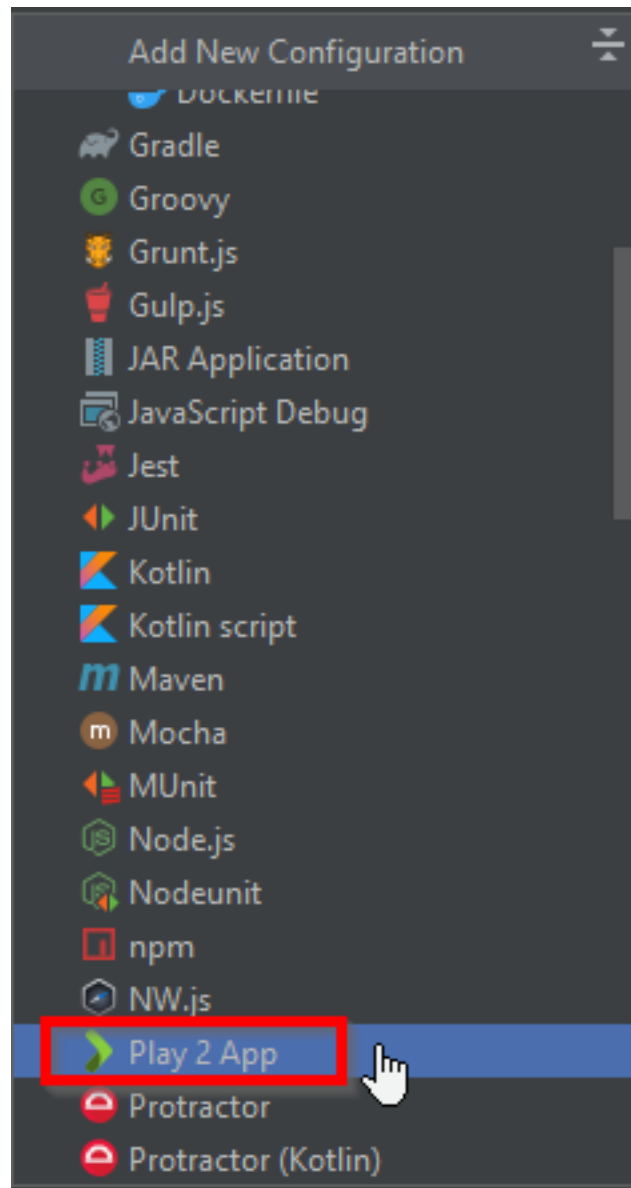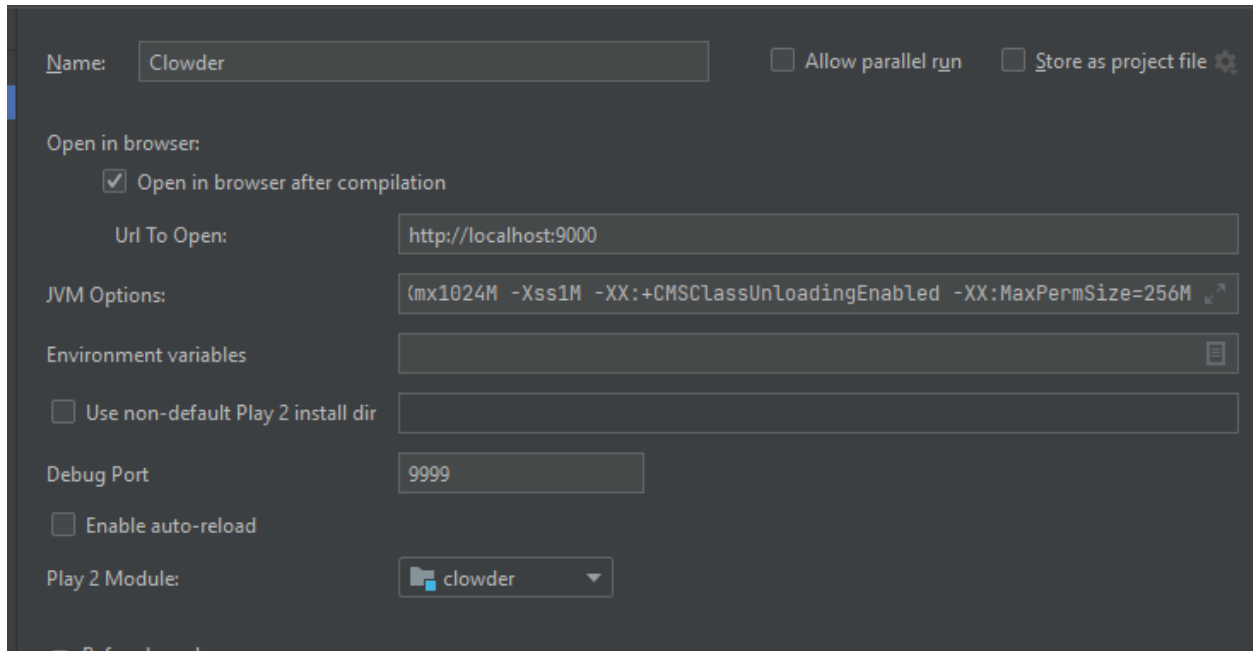
Fig. 1: The default Clowder run Configuration.

- FIRST_NAME

- LAST_NAME

- EMAIL_ADDRESS

- PASSWORD

- ADMIN (only set this if you want the user to have superadmin rights, make sure at lease one user has this).

Configuration complete! Now you can login to Clowder via `localhost:9000` in your browser.

> **Warning:** If you renamed the base clowder folder to something else, like *kitten*, then the `--netowrk` parameter must be changed to `--network kitten_clowder`.

Skip to using default extractors and developer resources

(Optional) User creation method 2: mock SMTP server

Enable local email verification

For local instances of Clowder, the email verification step will have to be done manually, via a mock SMTP email server.

Add the following lines to the bottom of `application.conf`:

```
# application.conf


# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Local email verification -- see Intellij's run console to complete registration
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
smtp.mock=true
```

All accounts must also be activated by an administrator. To activate your account by default, edit `application.conf`:

```
# application.conf
# Search for this line, and EDIT it (do not add a new line)
# Set to false

# Whether emails for new users registrations go through admins first
registerThroughAdmins=false
```

Now, create a local Clowder account via the web interface

Start Clowder:

1. Start required services (via `docker-compose up -d` from the root Clowder directory).

    1. You can check if your services are already running using `docker ps` and check that 3 containers are active (MongoDB, ElasticSearch, and RabbitMQ) by looking at `Server → Containser:  3`. Or check via the Docker Desktop GUI.

2. Ensure your local clowder instance is running (on `localhost:9000`)

Finally, **attempt to signup for an account via the Clowder GUI** on `localhost:9000`

- Click the Sign Up button in the top right.

Upon clicking Signup, **the IntelliJ console will show the text of the user signup verification emails**, where you can click the confirmation link.

Look for this in Intellij's run output terminal, **and click the link to complete registration**:

```
<p>Please follow this
    <a href="http://localhost:9000/signup/baf28c54-80fe-480c-b1e4-9200668cb92e">link</
↪a> to complete your registration
    at <a href="http://localhost:9000/">Clowder</a>.
</p>
```

- Don't see it? Make sure you enabled `smtp.mock=true` above.

Now fill in your account details, and you should be good to go using Clowder!

(Optional) Edit user properties directly in MongoDB

To edit the permissions on *existing accounts*, **edit their properties in MongoDB**. You can skip this step if you haven't created a local Clowder account yet.

1. Download a GUI for MongoDB: MongoDB Compass or a 3rd party tool like RoboMongo.

2. Ensure all services are running!

```
cd clowder # base directory

# start all required services
docker-compose up -d
```

1. Connect RoboMongo to the docker instance (the defaults should be fine)

    1. Point it towards port `27017`

2. To find user properties, in the file tree on the left, navigate to clowder → Collections → social.users

    1. Then click the dropdown to expand that user

---

2. Find `status` field, and right click to edit.

3. If it is `Inactive`, change it by typing `Active` (capitalized).

3. User is activated. Refresh your browser (on `localhost:9000`) to access Clowder.

If Clowder feels slow, add the faster JVM option

- Follow the instructions here to add JNotify.

- Simply download JNotify and tell IntelliJ where it is in the `Run Configurations` -> `JVM Options`.

1.5.4 Use the default extractors

The default extractors offer simple quality of life improvements for image, video, pdf, and audio file previews while browsing Clowder. Extractors also offer powerful capabilities for manipulating your data in Clowder, see PyClowder for additional capabilities, including running machine learning training on your data stored in Clowder.

Enable the default Extractors:

```
# start Clowder with extractor support
docker-compose -f docker-compose.yml -f docker-compose.override.yml -f docker-compose.
→extractors.yml up -d
```

Or run NCSA GeoServer for viewing and editing geospacial data via `docker-compose.geoserver.yml`:

- geoserver

- ncsa_geo_shp

- extractor-geotiff-preview

- extractor-geotiff-metadata

```
# start Clowder with default extractors, and GeoServer extractors
docker-compose -f docker-compose.yml -f docker-compose.override.yml -f docker-compose.
→extractors.yml -f docker-compose.geoserver.yml up -d
```

Learn more about GeoServer and read the documentation.

Troubleshooting extractors

First, make sure you include the extractors when starting Docker! See `docker-compose.extractors.yml` in the section above.

If running the extractor results in a `"Failed to establish a new connection: [Errno 111] Connection refused"`, this is a Docker networking issue. Containers must be able to talk to each other (Clowder talking to RabbitMQ).

To resolve, open `clowder/conf/application.conf` search for and set the RabbitMQ message queue URL:

```
clowder.rabbitmq.clowderurl="http://host.docker.internal:9000"
```

Simply saving the file should fix the issue. You can again "submit a file for extraction" on the file's details page. Done!

If navigating to `localhost:9000` yields nothing, try this. On Windows, I've had trouble getting `localhost` to resolve to the Docker host, so:

- Access Clowder **not** via localhost, but **via your local IP address**. For example, `55.251.130.193:9000`.

- **Find your local IP address:**

  - Windows: `Settings` -> `Network & internet` -> `IPv4 address`.

  - Mac: `System Preferences –> Netowrk–> Advanced–> TCP/IP–> IPv4 Address`. (Note: don't use the 'Public IP' from iStat Menus).

  - Linux `$ ifconfig`

That should resolve extractor issues across all major platforms, including Apple Silicon (M1).

1.5.5 Next Steps

Watch the Clowder Conference playlist on Youtube!

- 28 videos covering specific Clowder topics and uses

- Check out the Overview of Clowder

- Check out How to use Clowder

- Check out How to Create a New Extractor and many more!

Try the *default extractors* for simple quality of life improvements in Clowder.

```
docker-compose -f docker-compose.yml -f docker-compose.override.yml -f␣
↪docker-compose.extractors.yml up -d
```

Write your own extractors using the PyClowder Python package.

Please ask any questions on our Clowder Slack.

1.5.6 Resources for Developers

- Clowder REST API examples (and legacy examples)

- Customize your deployment

- Extractors: Running samples and writing your own

- Look at the Core Extractors for exmaples on how to use image, video, audio, PDF, etc.

  - Extractor for CSV files

  - Extractor for ZIP files

  - Virus checker extractor (to ensure datasets don't have viruses)

Can't find what you need? Clowder's legacy wiki may have additional detail.

1.5.7 Requirements Overview

Following is a list of requirements for the Clowder software. Besides Java, all other services/software can be installed on other machines with Clowder configured to communicate with them.

- Java 8 - required

  - The Clowder software is written in Scala and javascript and requires Java to execute.

  - Clowder has been tested with the OpenJDK.

  - Versions beyond 8 have not been tested.

- MongoDB v3.4 - required

    - By default Clowder uses MongoDB to store most of the information within the system.

    - Versions above 3.4 have not been tested.

- RabbitMQ (latest version) - optional

    - RabbitMQ is used to communicate between Clowder and the extractors. When deploying extractors it is required to deploy RabbitMQ as well.

- ElasticSearch 2.x - optional

    - ElasticSearch is used for text based search by Clowder.

    - Versions above 2.x have not been tested.

    - This dependency (specifically v2) is not compatible with Apple Silicon M1.

1.6 Web Service API

The RESTful application programming interface is the best way to interact with Clowder programmatically. Much of the web frontend and the extractors use this same API to interact with the system. For a full list of available endpoints please see the Swagger documentation.

Depending on the privacy settings of a specific Clowder instance, a user API key might be required. Users can create API keys through the web UI on their profile page (upper right user icon on any page). API Keys can be provided in the HTTP request as URL parameters, for example `?key=*yourapikey*` or using the HTTP header `X-API-Key: *yourapikey*`. The HTTP header method is preferred (more secure) but some environments / libraries might make it easer to provide the API key as a URL parameter.

You can use `curl` to test the service. If you are on Linux or MacOSX you should have it already. Try typing `curl` on the command prompt. If you are on windows, you can download a build at http://curl.haxx.se/. If you prefer more of a GUI experience, you can try Postman.

For example, the following examples request the metadata attached to a dataset.

Here is an example of requesting the metadata attached to a *dataset* and providing the API key as a URL parameter:

```
curl -X GET "https://clowderframework.org/clowder/api/datasets/
↪5cd47b055e0e57385688f788/metadata.jsonld?key=*yourapikey*"
```

Here is an example of requesting the metadata attached to a *file* and providing the API key as the HTTP header *X-API-Key*:

```
curl -X GET -H "X-API-Key: *yourapikey*" "https://clowderframework.org/clowder/api/
↪files/5d07b5fe5e0ec351d75ff064/metadata.jsonld"
```

1.7 Upgrading

This page describes how to upgrade the Clowder software. The steps described will do an in-place upgrade of Clowder. The biggest advantage of this upgrade is that it is fast and requires the least amount of changes to the current system.

1.7.1 Before you start

Before you start the upgrade thoroughly review the Change Log for all new versions since your current deployed version.

Confirm that your operating system, database, and other software installed still comply with the requirements for Clowder.

If you have installed Clowder extractors, verify that they will be compatible with the version of Clowder you are upgrading to. If not you will need to update the extractors as well.

We strongly recommend performing your upgrade in a test environment first. Do not upgrade your production Clowder server until you are satisfied that your test environment upgrade has been successful.

1.7.2 Backing up your database

Before you begin the upgrade process, make sure you have upgraded your database. During the upgrade process your database will be updated to match with the new version of the software. If you ever want to rollback to a previous version of the software you will have to rollback the database as well. Following are commands to backup your database, as well as the commands needed to restore the specific database

Backing up MongoDB

This will describe how to backup the mongo database. If you have the files stored in the mongo database (default) this can take a long time and take up a significant amount of space since it will also dump the actual files. This assumes you are using the default database name (clowder) on the local host. If your database is stored somewhere else or has a different name you will need to modify the commands below. To backup the mongo database use:

Listing 3: Backing up MongoDB

```
mongodump  --db clowder --out clowder-upgrade
```

Restoring MongoDB

This describes how to restore the mongo database. If you have the files stored in the mongo database (default) this can take a long time and take up a significant amount of space since it will also restore the actual files. There are two ways to restore the mongo database, the first one will drop the database first, and thus will also remove any additional collections you added. The second way will only drop those collections that are imported, this can leave some additional collections that could create trouble in future updates.

Listing 4: Restoring MongoDB 1

```
echo "db.dropDatabase();" | mongo --db clowder
mongorestore --db clowder clowder-upgrade/clowder
```

Listing 5: Restoring MongoDB 2

```
mongorestore --drop --db clowder clowder-upgrade/clowder
```

1.7.3 Performing the upgrade

The actual update consists of a few steps. After these steps are completed you will have an updated version of Clowder.

Make sure you have backed up your database.

Download the version you want to install, some common versions are:

- Latest stable version - This version is more tested, but is not as up to date as the development version.

- Latest development version - This version contains the latest code and has been lightly tested.

Stop the current version of Clowder you have running

Move the folder of the current version

Unzip the downloaded version of Clowder

Move the custom folder of the original Clowder to the custom folder of the new Clowder

Start Clowder. Make sure your startup script uses the flag *-DMONGOUPDATE=1* and *-DPOSTGRESUPDATE=1* to update the databases. If the database is not updated the application might not run correctly and/or you might not be able to login.

To make this process easier we have a script "update-clowder.sh" that will perform all these tasks for you (except for the backup, your are still responsible for the backup). The script does assume you have in the startup script that will have the UPDATE flags enabled.

To upgrade to the latest development version, as root, do:

```
CLOWDER_BRANCH=CATS-CORE0 ./update-clowder.sh
```

To upgrade to the latest stable version, as root, do:

```
./update-clowder.sh
```

For both, if this does not update it, add *–force* after *update-clowder.sh*.

1.7.4 Post upgrade checks and tasks

Once you have confirmed the availability of compatible versions of the extractors, you should upgrade your extractors after successfully upgrading Clowder.

Congratulations! You have completed your Clowder upgrade.

1.8 Customizing

1.8.1 The default configuration

> **Warning:** Do **not** make changes to the original files in `/conf`. Instead, create a `/custom` folder shown below.

The default configuration is fine for simple testing, but if you would like to modify any of the settings, you can find all the configuration files under the `/conf` directory. The following files are of particular importance:

- `/conf/application.conf` includes all the basic configuration entries. For example the MongoDB credentials for deployments where MongoDB has non default configuration.

- `/conf/play.plugins` is used to turn on and off specific functionality in the system. Plugins specific to Clowder are available under `/app/services`.

- `/conf/securesocial.conf` includes configuration settings for email functionality when signup as well as ways to configure the different identity providers (for example Twitter or Facebook). More information can be found on the securesocial website.

1.8.2 How to customize Clowder

To customize Clowder, create a folder called `custom` inside the Clowder folder (`clowder/custom`). Add the following. Modifications included in these files will overwrite defaults in `/conf/application.conf` and `/conf/play.plugins`.

```
cd clowder
mkdir custom
touch custom/application.conf custom/play.plugins
```

If you are working on the source code this folder is excluded from git so you can use that also to customize your development environment, and not accidentally commit changes to either `play.plugins` or `application.conf`. If you make any changes to the files in the custom folder you will need to *restart the application* (both in production and development).

play.plugins

The `/custom/play.plugins` file describes all the *additional* plugins that should be enabled. **This file can only add additional plugins, and is not capable of turning off any of the default ones enabled in** `/conf/play.plugins`. For example the following `play.plugins` file will enable some additional plugins:

Listing 6: play.plugins

```
9992:services.RabbitmqPlugin
10002:securesocial.core.providers.GoogleProvider
11002:services.ElasticsearchPlugin
```

custom.conf

/custom/custom.conf is used to override any of the defaults in the application.conf or any included conf files (such as securesocial.conf). Common changes we do is to modify Clowder to use a directory on disk to store all blobs instead of storing them in mongo. Following is an example that we use for some of the instances we have at NCSA.

- **One change every instance of Clowder should do is to modify the commKey and application.secret.**

Listing 7: custom.conf

```
# security options -- should be changed!
application.secret="some magic string"
commKey=magickey

# email when new user tries to sign up
smtp.from="no-reply@example.com"
smtp.fromName="NO REPLY"

# URL to mongo
mongodbURI = "mongodb://mongo1:27017,mongo2:27017,mongo3:27017/server1?
↪replicaSet=CLOWDER"

# where to store the blobs (highly recommended)
service.byteStorage=services.filesystem.DiskByteStorageService
medici2.diskStorage.path="/home/clowder/data"

# rabbitmq
clowder.rabbitmq.uri="amqp://user:password@rabbitmq/clowder"
clowder.rabbitmq.exchange=server1

initialAdmins="joe@example.com"

# elasticsearch
elasticsearchSettings.clusterName="clowder"
elasticsearchSettings.serverAddress="localhost"
elasticsearchSettings.serverPort=9300

# securesocial customization
# set this to true if using https
securesocial.ssl=true
# this will make the default timeout be 8 hours
securesocial.cookie.idleTimeoutInMinutes=480

# twitter setup
securesocial.twitter.requestTokenUrl="https://api.twitter.com/oauth/request_token"
securesocial.twitter.accessTokenUrl="https://api.twitter.com/oauth/access_token"
securesocial.twitter.authorizationUrl="https://api.twitter.com/oauth/authorize"
securesocial.twitter.consumerKey="key"
securesocial.twitter.consumerSecret="secret"
```

(continues on next page)

```
# google setup
securesocial.google.authorizationUrl="https://accounts.google.com/o/oauth2/auth"
securesocial.google.accessTokenUrl="https://accounts.google.com/o/oauth2/token"
securesocial.google.clientId="magic"
securesocial.google.clientSecret="magic"
securesocial.google.scope="https://www.googleapis.com/auth/userinfo.profile https://
→www.googleapis.com/auth/userinfo.email"

# enable cache
ehcacheplugin = enabled
```

messages.XY

This allows to translate or customize certain aspects of Clowder. All messages in Clowder are in english and are as messages.default. Unfortunately it is not possible to use messages.default to use for translations since it falls back to those embedded in the Clowder jar files. To update the messages in english, you can use messages.en. The default is for Clowder to only know about english, this can be changed in your custom.conf with `application.langs="nl"`.

Customizing Web UI

The `public` folder is place where you can place customizations for previews, as well as new stylesheets. To add a new stylesheet you should place it in the public/stylesheets/themes/ folder. The name should be <something>.min.css or <something>.css. The user will at this point see in their customization settings the option to select <something> as their new theme to be used.

To add new previews you can put them in the public/javascripts/previewers/. To create a previewer you will create a folder in there and in there have the files needed for the previewer as well as a package.json file. This package.json file will describe the previewer, which as the name, the main file to load, and the content types (Preview files) that the previewer can handle.

```
{
    "name" : "Video",
    "main" : "video.js",
    "contentType" : ["video/webm", "video/mp4", "video/videoalternativeslist"]
}
```

1.9 Architecture

Clowder's architecture consists of typical front end web application and several backend services. A Web user interface is provided out of the box for users interacting with the system through a Web browser. An extensive Web service API is provided for external clients to communicate with the system. These clients can include custom GUIs for specific use cases as well as headless scripts for system to system communication.

A service layer abstracts backend services so that individual deployments can be customized based on available resources. For example a user might want to store raw files on the file system, MongoDB GridFS, iRods or AWS S3 buckets.

When new data is added to the system, whether it is via the web interface or through the RESTful API, preprocessing is off-loaded to extraction services in charge of extracting appropriate data and metadata. The extraction services attempt to extract information and run preprocessing steps based on the type of the data just uploaded. Extracted information is then written back to the repository using appropriate API endpoints.
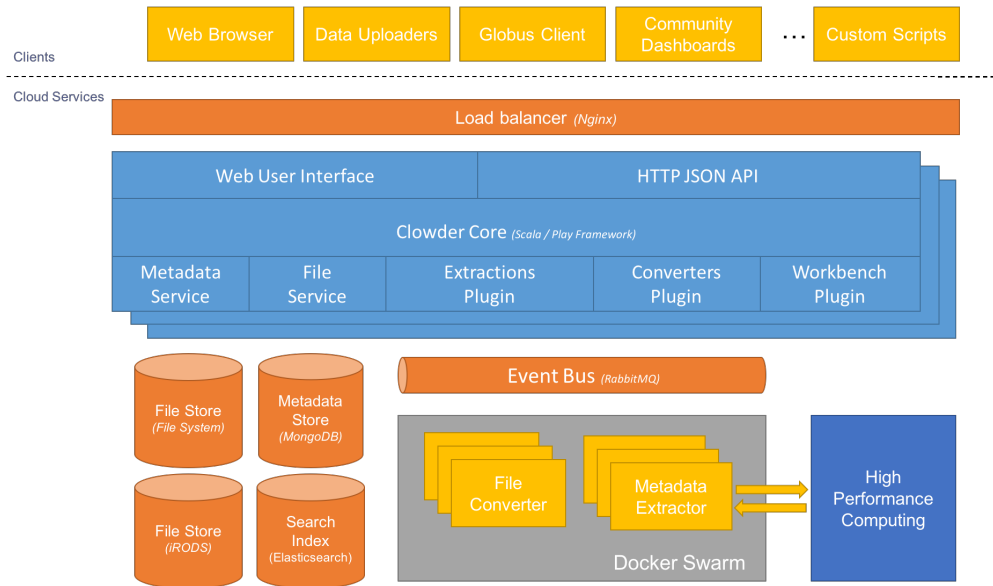
For example, in the case of images, a preprocessing step takes care of creating the previews of the image, but also of extracting EXIF and GPS metadata from the image. If GPS information is available, the web client shows the location of the dataset on a map embedded in the page. By making the clients and preprocessing steps independent the system can grow and adapt to different user communities and research domains.
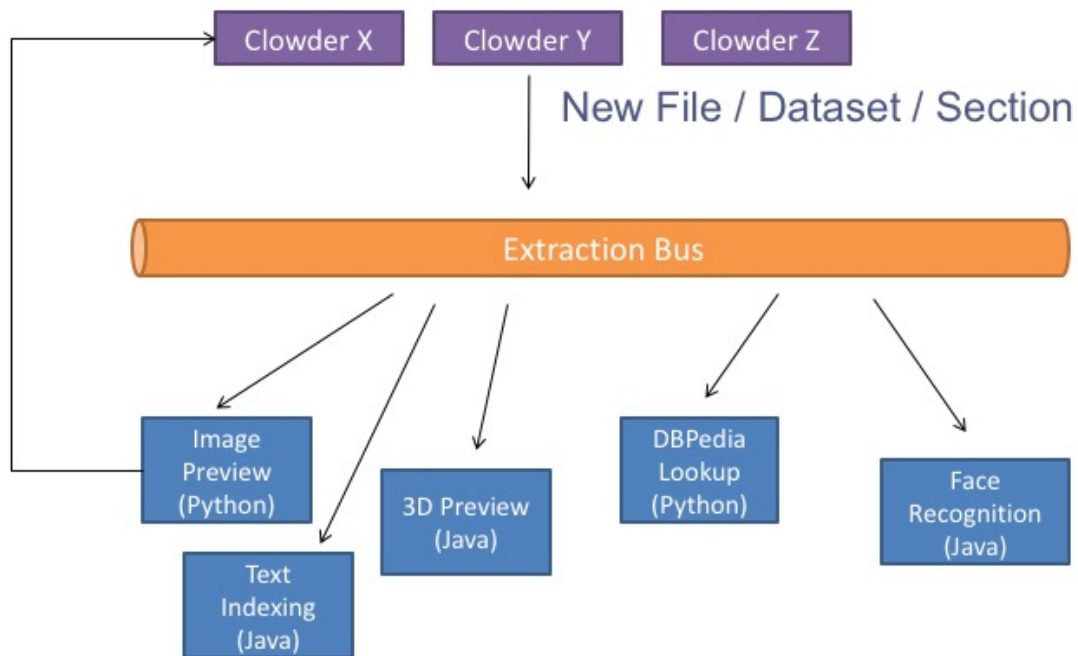
1.10 Extractors

- Overview
- Building and Deploying Extractors
- Testing Locally with Clowder
- A Quick Note on Debugging
- Additional pyClowder Examples

1.10.1 Overview

One of the major features of Clowder is the ability to deploy custom extractors for when files are uploaded to the system. A list of extractors is available in GitHub. A full list of extractors is available in Bitbucket.

To write new extractors, pyClowder is a good starting point. It provides a simple Python library to write new extractors in Python. Please see the sample extractors directory for examples. That being said, extractors can be written in any language that supports HTTP, JSON and AMQP (ideally a RabbitMQ client library is available for it).

Clowder capture several events and sends out specific messages for each. Extractors that are registered for a specific event type will receive the message and can then act on it. This is defined in the extractor manifest.

The current list of supported events is:

- File uploaded
- File added to dataset
- File Batch uploaded to dataset
- File remove from dataset
- Metadata added to file
- Metadata remove from file
- Metadata added to dataset
- Metadata removed from dataset
- File/Dataset manual submission to extractor

1.10.2 Building and Deploying Extractors

To create and deploy an extractor to your Clowder instance you'll need several pieces: user code, clowder wrapping code to help you integrate your code into clowder, an extractor metadata file, and, possibly, a Dockerfile for the deployment of your extractor. With these pieces in place, a user is able to search for their extractor, submit their extractor and have any metadata returned from their extractor stored - all within Clowder.

Although the main intent of an extractor is to interact with a file within Clowder and save metadata associated with said file, Clowder's ability to interact with files creates a flexibility with extractors that lets users do more than the intended scope. For instance, a user could write an extractor code that reads a file and pushes data to another application, modifies the file, or creates derived inputs within Clowder.

To learn more about extractor basics please refer to the following documentation.

For general API documentation refer here. API documentation for your particular instance of Clowder can be found under Help -> API.

1. User code

This is code written by you that takes, as input, a file(s) and returns metadata associated with the input file(s).

2. Clowder Code

We've created Clowder packages in Python and Java that make it easier for you to write extractors. These packages help wrap your code so that your extractor can be recognized and run within your Clowder instance. Details on building an extractor can be found at the following links:

- jClowder

- pyClowder

- From scratch using: * RabbitMQ client library * HTTP/JSON client libraries

3. extractor_info.json

The extractor_info.json file is a file that includes metadata about your extractor. It allows Clowder to "know" about your extractor. Refer here for more information on the extractor_info.json file.

4. Docker

To deploy your extractor within Clowder you need to create a Docker container. Docker packages your code with all its dependencies, allowing your code to be deployed and run on any system that has Docker installed. To learn more about Docker containers refer to docker.com. For a useful tutorial on Docker containers refer to katacoda.com. Installing docker requires a minimum of computer skills depending on the type of machine that you are using.

To see specific examples of Dockerfiles refer to the Clowder Code links above or peruse existing extractors at the following links:

- Clowder GitHub

- Clowder Bitbucket

If creating a simple Python extractor, a Dockerfile can be generated for you following the instructions on the clowder/generator) repository.

1.10.3 Testing Locally with Clowder

While building your extractor, it is useful to test it within a Clowder instance. Prior to deploying your extractor on development or production clusters, testing locally can help debug issues quickly. Below are some instructions on how to deploy a local instance of Clowder and deploy your extractor locally for quick testing. The following docker commands should be executed from a terminal window. These should work on a linux system with docker installed or on a mac and Windows with Docker Desktop) installed.

1. Build your docker image: run the following in the same directory as your Dockerfile

```
docker build -t myimage:tag .
```

2. Once your Docker image is built it can now be deployed within Clowder.

```
docker-compose -f docker-compose.yml -f docker-compose.extractors.yml up -d
```

Below are examples of each file:

- **docker-compose.yml**

    - This file sets up Clowder and its dependencies such as MongoDB and RabbitMQ. You should not have to modify it.

- **docker-compose.override.yml**

– This file overrides defaults, and can be used to customize clowder. When downloading the file, make sure to rename it to docker-compose.override.yml. In this case it will expose clowder, mongo and rabbitmq ports to the localhost.

- **docker-compose.extractor.yml**

– This file deploys your extractor to Clowder. You will have to update this file to reflect your extractor's name, Docker image name and version tag, and any other requirements like environment variables. See below:

```yaml
version: '3.5'

services:
  myextractor:
    image: myextractor_imagename:mytag
    restart: unless-stopped
    networks:
      - clowder
    depends_on:
      - rabbitmq
      - clowder
    environment:
      - RABBITMQ_URI=${RABBITMQ_URI:-amqp://guest:guest@rabbitmq/%2F}
      # Add any additional environment variables your code may need here
  # Add multiple extractors below following template above
```

1. Initialize Clowder. All the commands below assume that you are running this in a folder called tests, hence the network name tests_clowder. If you ran the docker-compose command in a folder called clowder, the network would be clowder_clowder.

```
docker run -ti --rm --network tests_clowder clowder/mongo-init
```

4. Enter email, first name, last name password, and admin: true when prompted.

5. Navigate to localhost:9000 and login with credentials you created in step 4.

6. Create a test space and dataset. Then click 'Select Files' and upload (if the file stays in CREATED and never moves to PROCESSED you might need to change the permission on the data folder using docker run -ti –rm –network tests_clowder clowder/mongo-init).

7. Click on file and type submit for extraction.

8. It may take a few minutes for you to be able to see the extractors available within Clowder.

9. Eventually you should see your extractor in the list and click submit.

10. Navigate back to file and click on metadata.

11. You should see your metadata present if all worked successfully.

1.10.4 A Quick Note on Debugging

To check the status of your extraction, navigate to the file within Clowder and click on the "Extractions" tab. This will give you a list of extractions that have been submitted. Any error messages will show up here if your extractor did not run successfully.

You can expand the tab to see all submissions of the extractor and any error messages associated with the submission:

If your extractor failed, the error message is not helpful, or if you do not see metadata present in the "Metadata" tab for the file you can check the logs of your extractor coming from the docker container by executing the following:

```
docker log tests_myextractor_1
```

Replace "myextractor" with whatever name you gave your extractor in the docker-compose.extractors.yml file.

If you want to watch the logs as your extractor is running you can type:

```
docker logs -f tests_myextractor_1
```

```
2021-08-16 16:39:32,050 [Thread-6        ] DEBUG   : pyclowder.connectors - [          edu']
2021-08-16 16:39:32,054 [Thread-6        ] INFO    : pyclowder.connectors - [6115383de4b01fd99b6ebbfb] : S
tatusMessage.start: Started processing.
2021-08-16 16:39:32,055 [Thread-6        ] DEBUG   : pyclowder.extractors - default check message : {'noti
fies': [          edu'], 'source': {'id': {'resourceType': "'file", 'id': '6115383de4b01fd99b6eb
bfb'}, 'mimeType': 'text/csv', 'extra': {'filename': 'test.csv'}}, 'jobid': '611a94c4e4b01fd99b6ebf2d', '
msgid': '611a94c4e4b01fd99b6ebf2e', 'flags': '', 'intermediateId': '6115383de4b01fd99b6ebbfb', 'host': 'h
ttp://clowder:9000', 'datasetId': '61002d8ee4b01fd99b6e9acb', 'id': '6115383de4b01fd99b6ebbfb', 'fileSize
': '31736', 'filename': 'test.csv', 'target': '{}', 'secretKey':
'activity': 'submitted', 'routing_key': 'extractors.ncsa.csvheaders', 'parameters': {}, 'action': 'manual
-submission'}
2021-08-16 16:39:32,082 [Thread-6        ] INFO    : pyclowder.connectors - [6115383de4b01fd99b6ebbfb] : S
tatusMessage.processing: Downloading file.
2021-08-16 16:39:32,111 [Thread-6        ] DEBUG   : __main__ - {'csv_headers': ['\ufeffEXTID', 'BARCD', '
PLTID', 'HNR', 'ABBRC', 'MATID']}
2021-08-16 16:39:32,112 [Thread-6        ] INFO    : pyclowder.connectors - [6115383de4b01fd99b6ebbfb] : S
tatusMessage.processing: Uploading file metadata.
2021-08-16 16:39:32,139 [Thread-6        ] INFO    : root - Uploaded metadata {'@context': ['https://clowd
er.ncsa.illinois.edu/contexts/metadata.jsonld', {'csv_headers': 'http://clowder.ncsa.illinois.edu/metadat
a/ncsa.csvheaders#csv_headers'}], 'attachedTo': {'resourceType': 'file', 'id': '6115383de4b01fd99b6ebbfb'
}, 'agent': {'@type': 'cat:extractor', 'extractor_id': 'http://clowder:9000/extractors/ncsa.csvheaders/1.
1', 'version': '1.1', 'name': 'ncsa.csvheaders'}, 'content': {'metadata': {'csv_headers': ['\ufeffEXTID',
 'BARCD', 'PLTID', 'HNR', 'ABBRC', 'MATID']}}}
2021-08-16 16:39:32,140 [Thread-6        ] INFO    : pyclowder.connectors - [6115383de4b01fd99b6ebbfb] : S
tatusMessage.done: Done processing.
```

You can print any debugging information within your extractor to the docker logs by utilizing the logging object within your code. The following example is for pyClowder:

```
logging.info("Uploaded metadata %s", metadata)
```

In the screenshot above you can see the lines printed out by the logging.info as the line will start with INFO:

```
2021-04-27 16:47:49,995 [MainThread     ] INFO
```

1.10.5 Additional pyClowder Examples

For a simple example of an extractor, please refer to extractor-csv. This extractor is submitted on a csv file and returns the headers as metadata.

Specifying multiple inputs

This example assumes data is within the same dataset.

```python
#!/usr/bin/env python3

import subprocess
import logging

from pyclowder.extractors import Extractor
import pyclowder.files
import pyclowder.datasets

class MyExtractor(Extractor):
        def __init__(self):
                Extractor.__init__(self)
                logging.getLogger('pyclowder').setLevel(logging.DEBUG)
                logging.getLogger('__main__').setLevel(logging.DEBUG)

                # Add an argument to pass second filename with default filename
                self.parser.add_argument('--secondfile',default="my_default_second_
→file.csv")
                self.setup()

        def process_message(self, connector,host, secret_key,resource, parameters):
                # grab inputfile path
                inputfile = resource["local_paths"][0]

                # get list of files in dataset
                filelist = pyclowder.datasets.get_file_list(connector, host, secret_
→key, parameters['datasetId'])

                # loop through dataset and grab id of file whose filename matches␣
→desired filename
```

(continues on next page)

```
                for file_dict in filelist:
                        if file_dict['filename'] == self.args.secondfile:
                                secondfileID = file_dict['id']

                # or a more pythonic way to do the above loop
                # secondfileId = [file_dict['id'] for file_dict in filelist if file_
→dict['filename'] == self.args.secondfile][0]

                # download second file "locally" so extractor can operate on it
                secondfilepath = pyclowder.files.download(connector, host, secret_key,
→ secondfileId)

                """
                Execute your function/code to operate on said inputfile and secondfile
                """

                # upload any metadata that code above outputs as "my_metadata"
                metadata = self.get_metadata(my_metadata, 'file', parameters['id'],
→host)
                pyclowder.files.upload_metadata(connector, host, secret_key,
→parameters['id'], metadata)

if __name__ == "__main__":
        extractor = MyExtractor()
        extractor.start()
```

Renaming files

```
class MyExtractor(Extractor):
        def __init__(self):
                Extractor.__init__(self)
                logging.getLogger('pyclowder').setLevel(logging.DEBUG)
                logging.getLogger('__main__').setLevel(logging.DEBUG)

                # Add an argument to pass second filename with default filename
                self.parser.add_argument('--filename')
                self.setup()

        def rename_file(self, connector, host, key, fileid,filename):
                # rename file
                renameFile= '%sapi/files/%s/filename' % (host, fileid)

                f = json.dumps({"name": filename})

                connector.put(renameFile,
                                      data=f,
                                      headers={"Content-Type": "application/json",
→"X-API-KEY": key},
                                      verify=connector.ssl_verify if connector else
→True)

        def process_message(self, connector, host, secret_key,resource, parameters):
                # grab inputfile path
                inputfile = resource["local_paths"][0]
```

```
            if self.args.filename:
                    # call rename_file function
                    self.rename_file(connector, host, secret_key, parameters['id
→'], self.args.filename)

                # upload any metadata that code above outputs as "my_metadata"
            metadata = self.get_metadata(my_metadata, 'file', parameters['id'],␣
→host)
            pyclowder.files.upload_metadata(connector, host, secret_key,␣
→parameters['id'], metadata)

if __name__ == "__main__":
        extractor = MyExtractor()
        extractor.start()
```

1.11 Previewers

Previewers are custom Javascript code to visualize information about datasets and files (collection support is experimental). Usually those are used to provide a preview of a file, when the file is big, but they can be used to visualize more interesting aspects of a resource. For example, the GIS previewers enable overlaying geospatial data on a interactive map in the browser.

Previewer can work together with extractors and external services.

Here is a list of previewer embedded with the core source.

1.12 How to Contribute Documentation

Documentation is stored in `doc/src/sphinx`.

Dependencies are stored in `doc/src/sphinx/requirements.txt`.

conda

Create a virtual environment for documentation:

```
conda create -n clowder_docs python=3.8 -y
conda activate clowder_docs
```

Now we must edit the *requirements.txt* file to be compatible with Conda. These packages are not available on conda-forge.

Comment out the top three lines like so:

```
# -i https://pypi.org/simple/
# sphinx-rtd-theme==0.5.0
# sphinx_design==0.0.13
...
```

Install the dependencies. It's always better to run all conda commands before installing pip packages.

```
conda install --file requirements.txt -y
pip install sphinx-rtd-theme==0.5.0 sphinx_design==0.0.13
```

pyenv

Create a virtual environment for documentation:

```
pyenv install 3.7.12 # or any 3.{7,8,9}
pyenv virtualenv 3.7.12 clowder_docs

# make virtual environemnt auto-activate
cd doc/src/sphinx
pyenv local clowder_docs
```

Install doc dependencies:

```
pip install -r requirements.txt
```

Now, build HTML docs for viewing:

```
# run from doc/src/sphinx
sphinx-autobuild . _build/html
```

Open http://127.0.0.1:8000 in your browser. Saved changes will be auto-updated in the browser.

(Optional alternative) Static builds

If you do not want dynamic builds, you can statically generate the HTML this way.

```
cd doc/src/sphinx
make html
```

View docs by opening `index.html` in the browser `clowder/doc/src/sphinx/_build/html/index.html`

If you experience *any* trouble, come ask us on Slack here!

Note: To see how to install Clowder, please see *Installing Clowder*.

1.13 Thank You

Atlassian for kindly providing an open source license to their software development products that make our daily efforts so much easier.

Balsamiq Mockups for kindly providing an open source license for their rapid wireframing tool that makes iterating over designs so much faster and enjoyable

A

C

D

E

F

H

L

O

P

S