

SKAO Datalake-as-a-Service Deployment

James Collinson

VRE WG 22/09/2022



Background

- SKA Regional Centres, **SRCs**
 - Turning data into science
- SKAO in the **ESCAPE** project
 - Prototypes of data orchestrator (**Rucio**) and science platform (**JupyterHub**) plus associated services
- SKAO **Rucio** (2021-)
 - Automated functional tests
 - Token-based auth functionality
- **JupyterHub** (2020-)
 - Environment generation (BinderHub)
 - Storage provisioning (shared/persistent user storage)

Transfers					
Src\Dst	SPSRC_STORM	IMPERIAL	CNAF	AUSSRC_STORM	STFC_STORM
STFC_STORM	100%	100%	100%	100%	NO DATA
SPSRC_STORM	NO DATA	100%	100%	100%	100%
IMPERIAL	100%	NO DATA	100%	100%	100%
CNAF	100%	100%	NO DATA	100%	100%
AUSSRC_STORM	100%	100%	100%	NO DATA	100%



SKAO use case testing on DLaaS ('DAC21', 2021)




The screenshot displays a JupyterLab environment with a web browser interface. The address bar shows the URL `https://escape-notebook.cern.ch/user/coll/lab/tree/data/sample_images`. The left sidebar contains a file browser for the `/ data / sample_images /` directory, listing various FITS files and their modification times. The main workspace is divided into three panes: a code editor for `1_prep.ipynb`, a code editor for `2_source_find.py`, and a terminal window labeled `Terminal 6`. The `2_source_find.py` pane shows a list of 14 islands with their coordinates and Gaussian fit flags. The terminal window displays the output of the script, including warnings about invalid keywords and a detailed summary of the image properties and island detection results.


```
Island #44015 (x=5057, y=1493): fit with 1 Gaussian with flag = 268
Island #44099 (x=5070, y=1990): fit with 1 Gaussian with flag = 256
Island #44125 (x=5071, y=670): fit with 1 Gaussian with flag = 268
Island #44176 (x=5082, y=92): fit with 1 Gaussian with flag = 256
Island #44270 (x=5092, y=327): fit with 1 Gaussian with flag = 258
Island #44361 (x=5109, y=0): fit with 1 Gaussian with flag = 266
Island #44399 (x=5110, y=2244): fit with 1 Gaussian with flag = 256
Island #44461 (x=5118, y=3795): fit with 1 Gaussian with flag = 256
Island #44589 (x=5136, y=1744): fit with 1 Gaussian with flag = 256
Island #44727 (x=5154, y=3425): fit with 1 Gaussian with flag = 256
Island #44732 (x=5154, y=847): fit with 1 Gaussian with flag = 256
Island #44734 (x=5156, y=1610): fit with 1 Gaussian with flag = 256
Island #45021 (x=5203, y=207): fit with 1 Gaussian with flag = 256
Island #45048 (x=5205, y=3199): fit with 1 Gaussian with flag = 278
Island #45062 (x=5200, y=890): fit with 1 Gaussian with flag = 256
Island #45080 (x=5205, y=1838): fit with 1 Gaussian with flag = 4
Island #45081 (x=5205, y=2095): fit with 1 Gaussian with flag = 4
Island #45084 (x=5205, y=1158): fit with 1 Gaussian with flag = 260
Island #45090 (x=5205, y=184): fit with 1 Gaussian with flag = 4
Island #45091 (x=5205, y=848): fit with 1 Gaussian with flag = 4
Island #45092 (x=5205, y=2709): fit with 1 Gaussian with flag = 286
Island #45093 (x=5205, y=2908): fit with 1 Gaussian with flag = 4
Island #45094 (x=5205, y=3367): fit with 1 Gaussian with flag = 4
Island #45095 (x=5205, y=3884): fit with 1 Gaussian with flag = 4
Island #45096 (x=5205, y=4242): fit with 1 Gaussian with flag = 4

Please check these islands. If they are valid islands and
should be fit, try adjusting the flagging options (use
show_fit with "ch0_flagged=True" to see the flagged Gaussians
and "help 'flagging_opts'" to see the meaning of the flags)
or enabling the wavelet module (with "atrous_do=True").
To include empty islands in output source catalogs, set
incl_empty=True in the write_catalog task.
WARNING: VerifyWarning: Invalid 'BLANK' keyword in header. The 'BLANK' keyword is only applicable to integer data, and will be ignored in this HDU. [astropy.io.fits.hdu.image]
WARNING: VerifyWarning: Invalid 'BLANK' keyword in header. The 'BLANK' keyword is only applicable to integer data, and will be ignored in this HDU.
--> Opened '1400mhz_1000h_pbcor.fits'
Image size ..... : (5204, 4776) pixels
Number of channels ..... : 1
Number of Stokes parameters ..... : 1
Beam shape (major, minor, pos angle) .... : (1.66667e-04, 1.66667e-04, 0.0) degrees
Frequency of image ..... : 1400.000 MHz
Number of blank pixels ..... : 0 (0.0%)
Flux from sum of (non-blank) pixels ..... : 0.037 Jy
--> Calculating background rms and mean images
/opt/conda/lib/python3.8/site-packages/numpy/core/fromnumeric.py:43: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
result = getattr(asarray(obj), method)(*args, **kwargs)
WARNING: Negative values found in rms map interpolated with spline_rank = 3
WARNING: Using spline_rank = 1 (bilinear interpolation) instead
Using user-specified rms box ..... : (74, 19) pixels
--> Using 2D map for background rms
--> Variation in mean image significant
--> Using 2D map for background mean
Min/max values of background rms map .... : (6.23e-08, 1.37e-05) Jy/beam
Min/max values of background mean map ... : (-6.50e-08, 9.69e-06) Jy/beam
Minimum number of pixels per island ..... : 6
Number of islands found ..... : 14812
Fitting islands with Gaussians ..... : [=====] 12395/14812
```



SKAO Data lake as a Service (2022)

- Deployed **StoRM-WebDAV** Rucio Storage Element (**RSE**) at STFC Cloud
- Enabled oversight of **all components** in Rucio stack (except FTS)
- Jupyter Notebook environments have read-access to StoRM-WebDAV RSE volume
- **Prerequisites:**
 - JupyterHub 
 - RSE with token auth 
 - Rucio server with token auth 

} colocated 

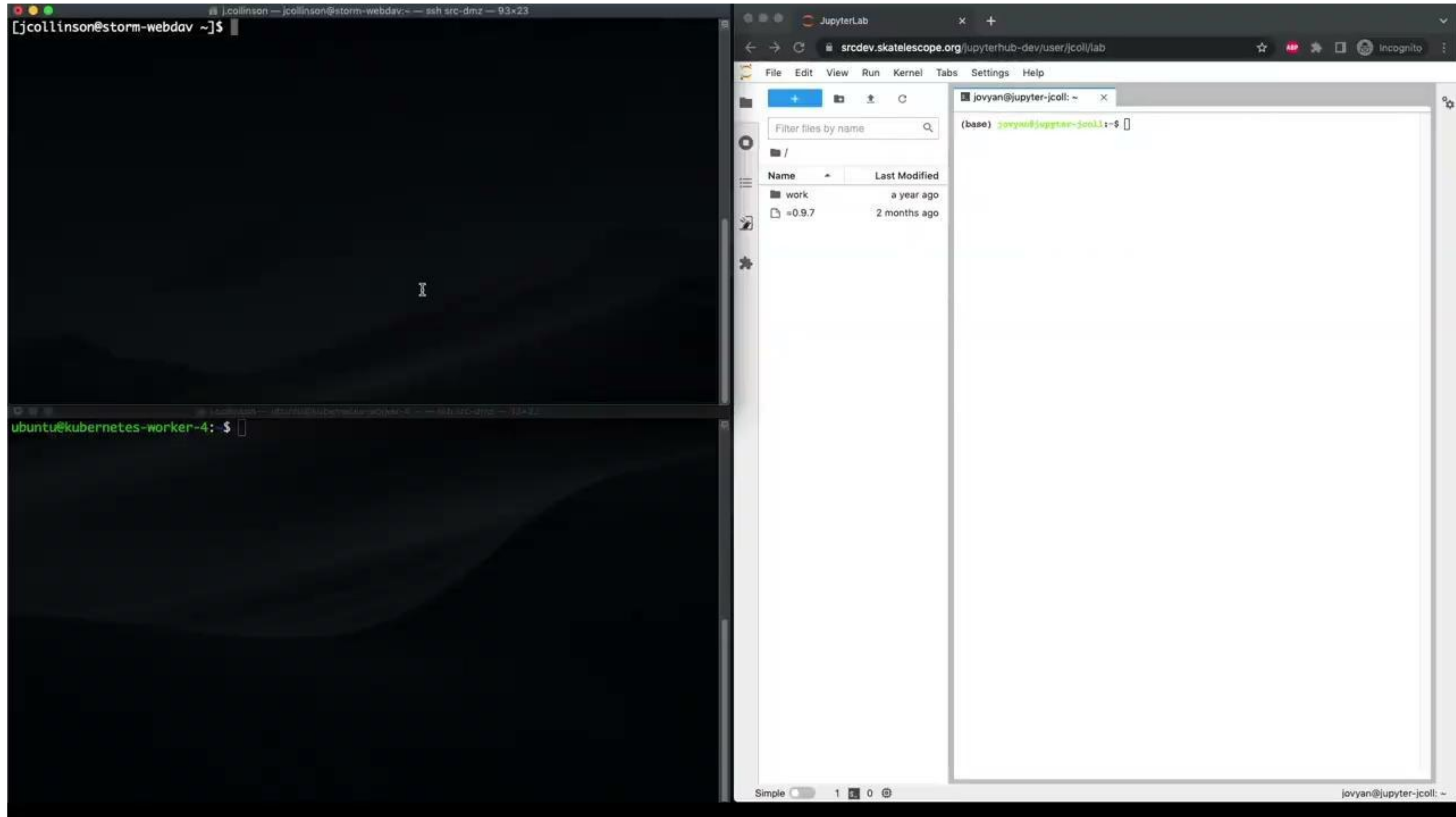


Deploying DLaaS (ESCAPE WP5 'Busy Week' Apr 22)

Data lake as a Service

- Developed by CERN as a means of integrating storage and compute
 - JupyterHub with extension to provide UI interaction with Rucio
 - Jupyter Notebook environments have read-access to local Rucio Storage Element (RSE) volume
 - Prerequisites to replicate:
 - JupyterHub ✓
 - RSE with token auth ✓
 - Rucio server with token auth ✓
- } colocated ✓

Deploying DLaaS (ESCAPE WP5 'Busy Week' Jul 22)



Value

- Technical demo prototyping user interaction with Rucio
 - **Auth** is most valuable component, as one of the most complex aspects of Rucio generally
- Deployment/**operations** experience was key
- Ideal testbed would be a future SKAO **science data challenge**
 - Currently looking at using Rucio for (part of) data distribution for SDC3A (Q4 2022)
 - Provide the closest analogy to **SRC user community** in next 2-3 years



Issues

- Uploads

- Currently not working via UI - complex set up on ESCAPE instance to workaround initial lack of token-based uploads?
- Should be possible to directly use Rucio CLI in rucio-jupyterlab extension now

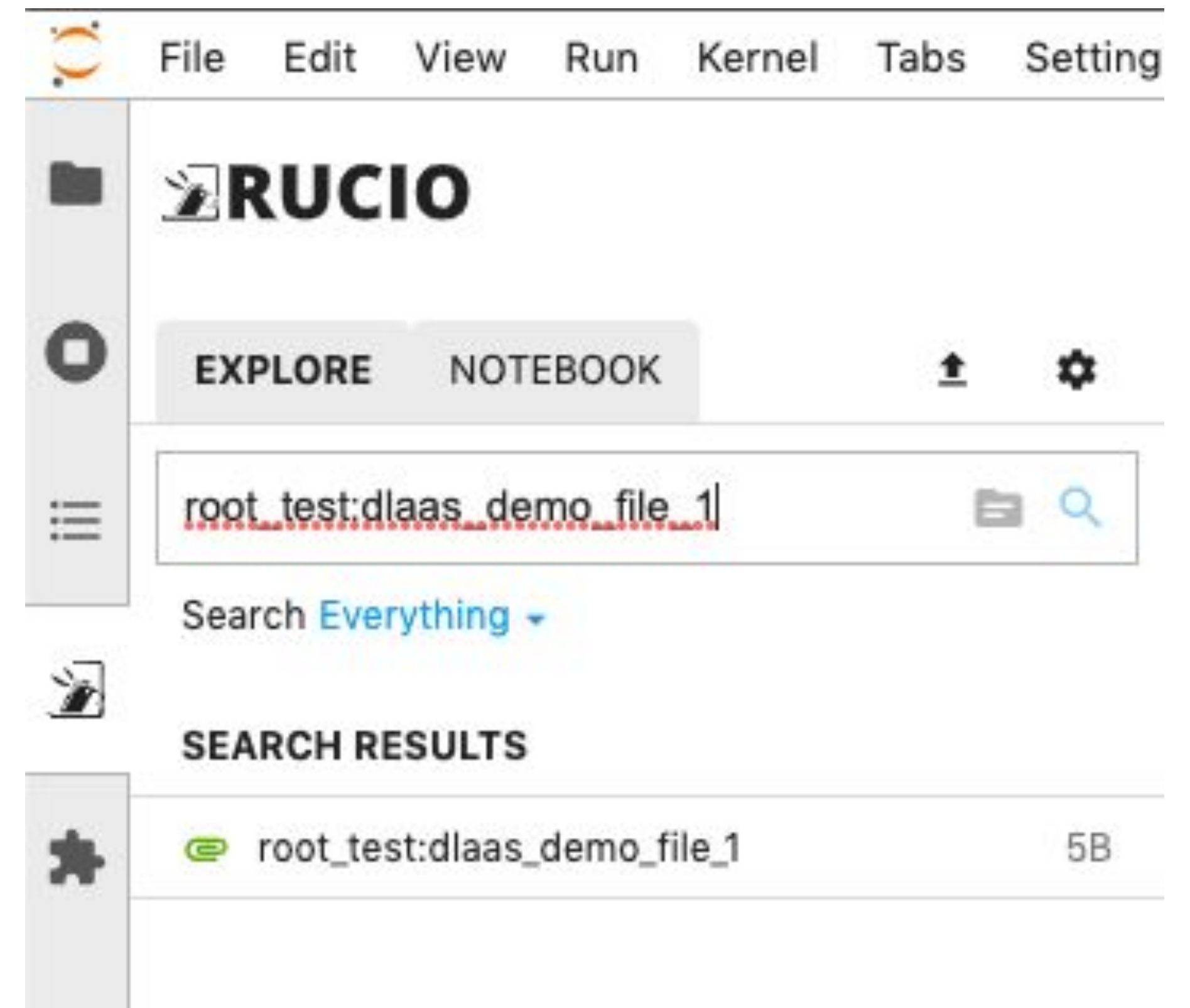
- Token refresh

- Alba/Elena encountered this too - in most recent version of Indigo IAM, `offline_access` scope prohibited for token exchange, where source and target IAM clients are the same
- Means refresh token cannot be requested, user must re-login after 1 hr



Higher level

- User friendly?
 - User ideally shouldn't need to know about Rucio or how data is managed behind the scenes
- Limitation as to how user friendly interface which requires DID-based querying can be; more useful (SKAO use case) to allow query by metadata?



Next steps

- Have made notes on internal Confluence for others in SRC prototyping teams to follow
 - Manuel Parra (IAA) currently working through and creating improved public docs relating to Spain SRC deployment
- Where would be best place for longer-lived (public) deployment docs to be?
- As this is my first VRE WG, interested to learn other groups' plans (if any) to further develop



Video Links

1) SKAO workflow demo on ESCAPE DLaaS instance:

<https://drive.google.com/file/d/1nDDtbXipQeZBiDtN7GVp71NlfVqxYhga/view?usp=sharing>

2) Deploying DLaaS step 1 - configuring storage:

<https://drive.google.com/file/d/1Q8qNcAKPTRVzMt77ZM0Vofxq0iikjFI2/view?usp=sharing>

3) Deploying DLaaS step 2 - configuring Rucio-JupyterLab extension and authentication:

<https://drive.google.com/file/d/1oYFlqK4RxkUrNIO6cYFjD9TyOzk9YvD1/view?usp=sharing>

