Scripted Data Transfers – converge to Roar Collab

Set up Service Account:

Username: other 996be69de6544b

Service Account Service Response

Service Account Password *NOTE* Save for later use. Do Not Lose!



Guest Collections:

converge:

/storage/long/

Roar Collab:

/storage/group/MCL/default/globus-share/



MRI_Converge_Guest_Collection
Guest Collection (GCS) on PennState_MRI_Converge



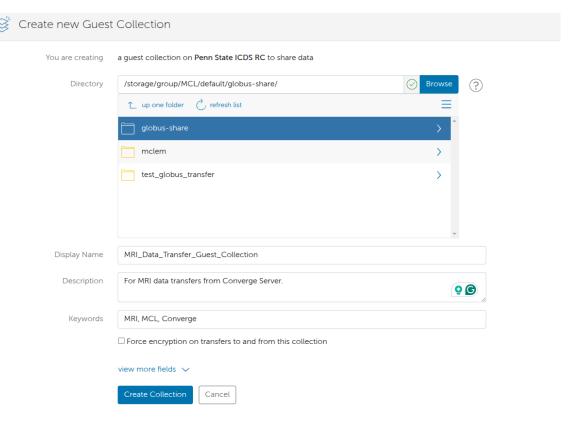
MRI_Data_Transfer_Guest_Collection

Guest Collection (GCS) on Penn State ICDS RC

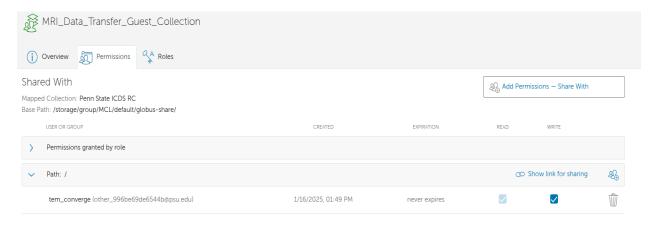
Creating Globus Guest Collections:

Go to https://www.globus.org/ and Log In

Click "Collections" > Under Collection, click "PennState_MRI_Converge" > and the in the Collections Tab click "Add Guest Collection"



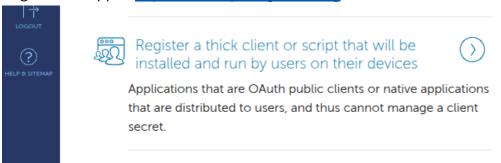
Grant Service Account RW Permission:



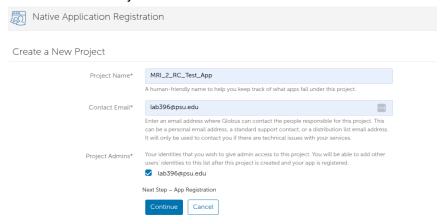
Developing an Application Using Globus Auth -

https://docs.globus.org/cli/examples/

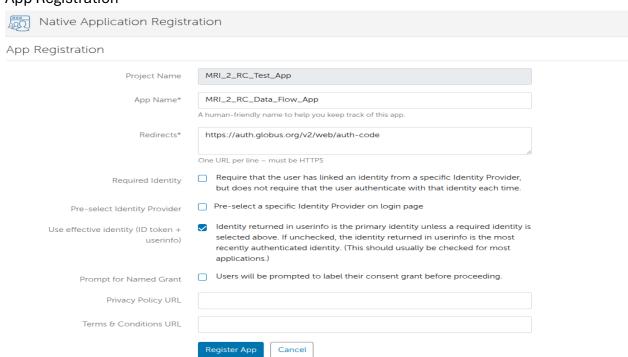
Register the App at https://developers.globus.org



Create a New Project

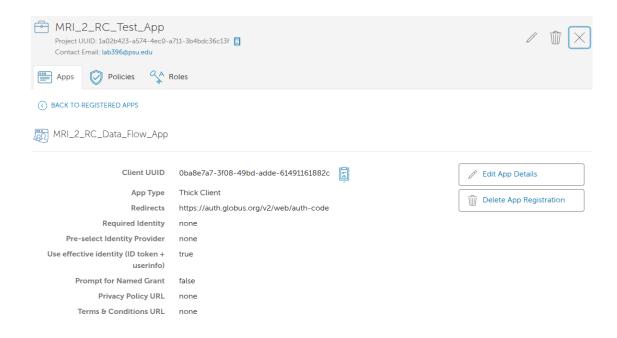


App Registration



Click "Register App" and the following screen will appear after registration:

NOTE: UUID - 0ba8e7a7-3f08-49bd-adde-61491161882c



The created registered application is now used in the next step to provide for the automated flows using persistent authentication. The application will be referenced via the field "Client UUID" and is highlighted in the above image.



The above section is a minimal set of instructions to create a working registered app for use with automated flows. For further information, please reference the Globus application developers guide available from URL https://docs.globus.org/api/auth/developer-guide

Authentication via the use of a Refresh Token

Prerequisites:

- 1. Anaconda3/Python
- 2. Globus-SDK
- 3. Python script(s)

I adapted a simple "ls" python script to test the globus auth and app.

Snippet of my code for "ls" python script

```
from globus_sdk import GroupsClient, NativeAppAuthClient, RefreshTokenAuthorizer
from globus_sdk.tokenstorage import SimpleJSONFileAdapter
 For this example, this is the app "My Data Flow Application", created in the earlier section.
auth_client = NativeAppAuthClient(CLIENT_ID)
ay_file_adapter = SimpleJSONFileAdapter(
    # instead of the defaults, only request access to the Transfer API auth_client.oauth2_start_flow(requested_scopes=TransferScopes.all, refresh_tokens=True)
    print(f"Native App Authorizaion URL:\n{authorize_url}\n")
```

Run "ls.py" to test auth:

/opt/anaconda3/bin/python ls.py

```
root@converge:-/globus_auth_scripts# /opt/anaconda3/bin/python --version
Python 3.12.7
```

Go to URL in output:



MRI_2_RC_Data_Flow_App would like to:



Manage data using Globus Transfer (j)



To work, the above will need to: ~

By clicking "Allow", you allow MRI_2_RC_Data_Flow_App (this client has not provided terms of service or a privacy policy to Globus) to use the above listed information and services. You can rescind this and other consents 🖾 at any time.



Deny

Click "Allow":



Native App Authorization Code

VsAuKZTklqWSaUhLGI8pHcyDvpdrqk

Authorization codes are valid for 10 minutes: this code will expire at 8:43 AM.

This authorization code may be revoked from the Manage Consents page.

Copy the Authorization Code, and head back to your terminal.

```
root@converge:~/globus_auth_scripts# /opt/anaconda3/bin/pytNative App Authorizaion URL:
https://auth.globus.org/v2/oauth2/authorize?client_id=0ba8eth-code&scope=urn%3Aglobus%3Aauth%3Ascope%3Atransfer.api.glYGnusv7GZQJfhwUas&code_challenge_method=S256&access_type=ofPlease enter the code here: VsAuKZTklqWSaUhLGI8pHcyDvpdrqk
```

If all works, you should see the output of the simple 'ls' test:

```
Please enter the code here: VsAuKZTklqWSaUhLGI8pHcyDvpdrqk
dir aks6717 2 2750 username umg/up.icds.rc.em.aks6717 2024-02-14 19:36:40+00:00
dir atm2 2 2750 username umg/up.icds.rc.em.atm2 2024-02-14 19:36:40+00:00
dir bkc5474 2 2750 username umg/up.icds.rc.em.bbc5474 2024-02-14 19:36:40+00:00
dir bul14 2 2750 username umg/up.icds.rc.em.bul14 2024-02-14 19:36:40+00:00
dir dew125 2 2750 username umg/up.icds.rc.em.dew125 2024-02-14 19:36:40+00:00
dir dfy55267 3 2750 username umg/up.icds.rc.em.defx5267 2024-10-25 20:20:00+00:00
dir dfp5170 2 2750 username umg/up.icds.rc.em.dfp5170 2024-02-14 19:36:40+00:00
dir dfp5170 2 2750 username umg/up.icds.rc.em.dfp5170 2024-02-14 19:36:40+00:00
dir dfp5170 2 2750 username umg/up.icds.rc.em.dfp23 2024-02-14 19:36:40+00:00
dir gsa5089 2 2750 username umg/up.icds.rc.em.gsa5089 2024-02-14 19:36:40+00:00
dir jlf500 2 2750 username umg/up.icds.rc.em.jf500 2024-02-14 19:36:40+00:00
dir jlf375 3 2750 username umg/up.icds.rc.em.jf500 2024-02-14 19:36:40+00:00
dir jlf375 3 2750 username umg/up.icds.rc.em.jp375 2024-12-19 18:31:58+00:00
dir jsa22 2 2750 username umg/up.icds.rc.em.jsa249 2024-02-14 19:36:40+00:00
dir jza449 2 2750 username umg/up.icds.rc.em.jsa249 2024-02-14 19:36:40+00:00
dir jza459 2 2750 username umg/up.icds.rc.em.jsa249 2024-02-14 19:36:40+00:00
dir maf43 2 2750 username umg/up.icds.rc.em.maf43 2024-02-14 19:36:40+00:00
dir maf43 2 2750 username umg/up.icds.rc.em.maf43 2024-02-14 19:36:40+00:00
dir maf536 2 2750 username umg/up.icds.rc.em.maf3 2024-02-14 19:36:40+00:00
dir maf54 2 2750 username umg/up.icds.rc.em.maf3 2024-02-14 19:36:40+00:00
dir njs6356 2 2750 username umg/up.icds.rc.em.maf3 2024-02-14 19:36:40+00:00
dir res20 2 2750 username umg/up.icds.rc.em.rip6365 2024-02-14 19:36:40+00:00
dir rip64 2 2750 username umg/up.icds.rc.em.rip64 2024-02-14 19:36:40+00:00
dir ruv12 2 2750 username umg/up.icds.rc.em.rip64 2024-02-14 19:36:40+00:00
dir staff 14 2750 username umg/up.icds.rc.em.sx530 2024-02-14 19:36:40+00:00
dir sxx530 2 2750 username umg/up.icds.rc.em.sx5
```

And, the generation of your "my-globus-refresh-token.json":

```
root@converge:~/globus_auth_scripts# ls
ls.py my-globus-refresh-token.json
```

NOTES:

ADD SERVICE ACCOUNT TO mcl collab

lab396@submit02 ~]\$ rcusers atg mcl_collab other_996be69de6544b Adding 'other_996be69de6544b' to 'H6-mcl_collab'