

# Data transfers using http endpoints

## Proxies

At INFN-Tier-1, valid WebDAV endpoints for the experiments' storage areas are provided with StoRM WebDAV (third-party-copy supported) or Apache. The public page with all the features to properly contact a CNAF endpoint is available at [https://www.cnaf.infn.it/~usersupport/Webdav\\_SA.html](https://www.cnaf.infn.it/~usersupport/Webdav_SA.html).

Then, the most common WebDAV clients can be used to access the storage areas, namely browsers and command-line tools such as `curl` and `gfal`.

When StoRM WebDAV is used, VOMS proxies are supported only by command-line tool, and browsers can be used to navigate into the storage area content if anonymous read-only access is enabled (HTTP endpoint) or if VO users access by their X509 certificate is enabled (HTTPS endpoint).

A few useful commands follow and more info are available at [\[19\]](#).

With a valid voms-proxy:

```
[arendina@ui-tier1 ~]$ voms-proxy-init --voms juno
Contacting lcgvoms02.jinr.ru:15008 [/C=RU/O=RDIG/OU=hosts/OU=jinr.ru/CN=lcgvoms02.jinr.ru] "juno"...
Remote VOMS server contacted succesfully.
Created proxy in /tmp/x509up_u10162.
Your proxy is valid until Fri Jul 02 05:42:21 CEST 2021

[arendina@ui-tier1 ~]$ voms-proxy-info --all
subject   : /DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Andrea Rendina
arendina@inf.n.it/CN=2090475310
issuer    : /DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Andrea Rendina
arendina@inf.n.it
identity  : /DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Andrea Rendina
arendina@inf.n.it
type      : RFC3820 compliant impersonation proxy
strength  : 2048
path      : /tmp/x509up_u10162
timeleft  : 11:59:35
key usage : Digital Signature, Key Encipherment
=== VO juno extension information ===
VO        : juno
subject   : /DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Andrea Rendina
arendina@inf.n.it
issuer    : /C=RU/O=RDIG/OU=hosts/OU=jinr.ru/CN=lcgvoms02.jinr.ru
attribute : /juno/Role=NULL/Capability=NULL
timeleft  : 11:59:35
uri       : lcgvoms02.jinr.ru:15008
```

- **Listing directory**

```
[arendina@ui-tier1 ~]$ gfal-ls davs://xfer-archive.cr.cnaf.infn.it:8443/juno
rucio4juno
cronTest
```

- **Upload**

```
[arendina@ui-tier1 ~]$ gfal-copy test.txt davs://xfer-archive.cr.cnaf.infn.it:8443/juno/test0107
Copying file:///home/USER-SUPPORT/arendina/test.txt [DONE] after 0s
```

- **Download**

```
[arendina@ui-tier1 ~]$ gfal-copy davs://xfer-archive.cr.cnaf.infn.it:8443/juno/test0107 here
Copying davs://xfer-archive.cr.cnaf.infn.it:8443/juno/test0107 [DONE] after 0s
```

- **Removing a file**

```
[arendina@ui-tier1 ~]$ gfal-rm davs://xfer-archive.cr.cnaf.infn.it:8443/juno/test0107
davs://xfer-archive.cr.cnaf.infn.it:8443/juno/test0107  DELETED
```

## Third-party-copies

In order to properly perform a third-party-copy between two endpoints which support the [http](#) protocol it occurs to set the BEARER\_TOKEN up, also known as macaroon.

Indeed, this token is used to authenticate the user always to the second endpoint. For this reason, the server to contact to get the token depends on whether the copy is in pull or push mode.

Actually, if both the endpoints are able to release a macaroon and the used gfal version is greater or equal to the 2.20 [\[28\]](#), the process is automatic. Otherwise, if only one of the two endpoints can release a BEARER\_TOKEN, or equivalently just one endpoint supports the http protocol, the user must request a macaroon to that endpoint.

Two easy examples follow below.

### Pull-copy

The user has to authenticate itself with a valid voms-proxy to the first endpoint, in this case IHEP, and then IHEP will authenticate itself with a BEARER\_TOKEN that the user has got previously from the second endpoint, namely the CNAF server (this happens because a pull-copy triggers a GET request). So, step by step:

- I ask a BEARER\_TOKEN to the second endpoint, namely "xfer-archive" at CNAF.

```
-bash-4.2$ export BEARER_TOKEN=$(curl -s --cacert ~/.globus/usercert.pem --cert $X509_USER_PROXY --key $X509_USER_PROXY --capath /etc/grid-security/certificates/ -X POST -d grant_type=client_credentials https://xfer-archive.cr.cnaf.infn.it:8443/oauth/token | jq -r .access_token)
```

- I can pull a file from CNAF to IHEP.

```
-bash-4.2$ gfal-copy -v --copy-mode pull davs://xfer-archive.cr.cnaf.infn.it:8443/juno/test_Andrea davs://junoeos01.ihep.ac.cn:9000/eos/juno/user/rucio_test/test_Andrea_big
```

### Push-copy

On the other hand, since the push-copy triggers a COPY request, the second endpoint to contact is CNAF again.

- I ask a BEARER\_TOKEN to the second endpoint, as above.

```
-bash-4.2$ export BEARER_TOKEN=$(curl -s --cacert ~/.globus/usercert.pem --cert $X509_USER_PROXY --key $X509_USER_PROXY --capath /etc/grid-security/certificates/ -X POST -d grant_type=client_credentials https://xfer-archive.cr.cnaf.infn.it:8443/oauth/token | jq -r .access_token)
```

- Then, I can push a file from IHEP to CNAF.

```
-bash-4.2$ gfal-copy -v --copy-mode push davs://junoeos01.ihep.ac.cn:9000/eos/juno/user/rucio_test/test_Andrea_big davs://xfer-archive.cr.cnaf.infn.it:8443/juno/test_to_CNAF
```

## Tokens

StoRM WebDAV also supports OpenID connect authentication and authorization on storage areas, so tokens can be used instead of proxies [\[20\]](#). Dedicated IAM (Identity and Access Management) instances can be configured for the experiments upon requests (please contact the user support). The list of the storage areas divided into the proper token-issuers is available at the following link: [https://www.cnaf.infn.it/~usersupport/Webdav\\_token.html](https://www.cnaf.infn.it/~usersupport/Webdav_token.html).

Also, a catch-all IAM instance is available at [iam-t1-computing.cloud.cnaf.infn.it](https://iam-t1-computing.cloud.cnaf.infn.it) where specific groups can be created for different communities, and users can register and request group membership. In this case, exploiting the support for fine-grained authorization within StoRM WebDAV, users can browse their data after OIDC login, and/or they can access storage from command line using `gfal` ([Gfal-utils](#)) or `curl` with JWT tokens.

Once registered within IAM, an access token can be retrieved using [OIDC-agent](#), which is available in ui-tier1. A full set of commands follows below.

At first, we have to start the process with the command

```
[arendina@ui-tier1 ~]$ eval `oidc-agent-service use`  
Agent pid 20578
```

Then, one needs to register his own client ([this has to be done just the first time to create a new local IAM client](#)):

```
[arendina@ui-tier1 ~]$ oidc-gen -w device
```

following the steps required by the application, and namely:

- initialize his OIDC client's name
- insert IAM URL (e.g. <https://iam-t1-computing.cloud.cnaf.infn.it/>)
- when a web page for authentication opens, enter the credentials of the IAM account
- authorize the new client
- insert your client's password twice.

Whereas, from the second time it is enough typing:

```
[arendina@ui-tier1 ~]$ oidc-add <your-client-name>  
Enter decryption password for account config '<your-client-name>':  
success
```

To get the access token and save it an environment variable:

```
[arendina@ui-tier1 ~]$ TOKEN=$(oidc-token <your-client-name>)
```

The token will be valid for 60 minutes. It can be obtained a new token issuing the oidc-token command again.  
At the end, stop oidc-agent daemon:

```
oidc-agent-service stop
```

Moreover, to properly use the gfal tools it occurs to set the BEARER\_TOKEN environment variable up.

A useful full set of commands can be found here:

```
-bash-4.2$ eval `oidc-agent-service use`  
Agent pid 17216  
-bash-4.2$ oidc-add juno2  
Enter decryption password for account config 'juno2':  
success  
-bash-4.2$ export BEARER_TOKEN=$(oidc-token juno2)  
-bash-4.2$ gfal-ls davs://xfer-archive.cr.cnaf.infn.it:8443/juno/  
rucio4juno  
test_Andrea  
cronTest
```

NB: gfal first tries to access the resource with the user X509 voms-proxy and this could lead to errors.  
In order to avoid this, the user must run **voms-proxy-destroy** to properly switch to the JWT authentication.

Furthermore, an error like this could appear, but it is just an irrelevant warning message:

```
(Davix::OpenSSL) Error: impossible to open /tmp/x509up_u10164: : error:02001002:system library:fopen:No such  
file or directory
```

On the other hand, to switch back to the voms-proxy it occurs to unset the BEARER\_TOKEN, namely:

```
-bash-4.2$ unset BEARER_TOKEN

-bash-4.2$ voms-proxy-init --voms belle
Contacting voms.cc.kek.jp:15020 [/C=JP/O=KEK/OU=CRC/CN=host/voms.cc.kek.jp] "belle"...
Remote VOMS server contacted succesfully.

Created proxy in /tmp/x509up_ucopy test https://junoeos01.ihep.ac.cn:9000/eos/juno/dirac/test_now

-bash-4.2$ gfal-ls davs://xfer-archive.cr.cnaf.infn.it:8443/belle
bellehttpd
TMP
CONTENT.stats
rucio4iddls
```

## Curl examples

A user can also use the curl command to make data management with the StoRM WebDAV storage areas and tokens. As shown before, some steps to retrieve a valid token are needed:

```
-bash-4.2$ eval `oidc-agent-service use`
25684
-bash-4.2$ oidc-add juno
Enter decryption password for account config 'juno':
success
-bash-4.2$ export BEARER_TOKEN=$(oidc-token juno)
```

The same operations performed using the gfal utils can be reproduced also with the curl command. A few useful examples follow below:

- **Listing directory**

```
-bash-4.2$ curl -H "Authorization: Bearer $BEARER_TOKEN" --capath /etc/grid-security/certificates/
https://xfer-archive.cr.cnaf.infn.it:8443/juno-test/
```

- **Upload**

```
-bash-4.2$ curl --capath /etc/grid-security/certificates -H "Content-Type: text/csv" -H "Authorization:
Bearer $BEARER_TOKEN" -X PUT https://xfer-archive.cr.cnaf.infn.it:8443/juno-test/test_Andrea --data-
binary "@test"
```

where test is the file in the user local folder.

- **Download**

```
-bash-4.2$ curl --capath /etc/grid-security/certificates -H "Content-Type: text/csv" -H "Authorization:
Bearer $BEARER_TOKEN" https://xfer-archive.cr.cnaf.infn.it:8443/juno-test/test_Andrea -o local_copy
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100    10  100    10    0    0    22    0  --:--:--  --:--:--  --:--:--    22 (edited)
```

- **Create a directory**

```
-bash-4.2$ curl --capath /etc/grid-security/certificates -H "Authorization: Bearer $BEARER_TOKEN" -X
MKCOL https://xfer-archive.cr.cnaf.infn.it:8443/juno-test/test_dir
```

- **Removing a file or a directory**

```
-bash-4.2$ curl --capath /etc/grid-security/certificates -H "Content-Type: text/csv" -H "Authorization:
Bearer $TOKEN" -X DELETE https://xfer-archive.cr.cnaf.infn.it:8443/juno-test/test_dir
```

```
[fornariclas12@ui-tier1 ~]$ eval `oidc-agent-service use`  
Agent pid 24806  
  
[fornariclas12@ui-tier1 ~]$ oidc-gen  
Enter short name for the account to configure: fornariclas12  
[1] https://iam-test.indigo-datacloud.eu/  
[2] https://iam.deep-hybrid-datacloud.eu/  
[3] https://iam.extreme-datacloud.eu/  
[4] https://b2access.eudat.eu/oauth2/  
[5] https://b2access-integration.fz-juelich.de/oauth2  
[6] https://unity.eudat-aai.fz-juelich.de/oauth2/  
[7] https://unity.helmholtz-data-federation.de/oauth2/  
[8] https://login.helmholtz-data-federation.de/oauth2/  
[9] https://services.humanbrainproject.eu/oidc/  
[10] https://accounts.google.com/  
[11] https://aai.egi.eu/oidc/  
[12] https://aai-dev.egi.eu/oidc  
[13] https://login.elixir-czech.org/oidc/  
[14] https://oidc.scc.kit.edu/auth/realms/kite/  
Issuer [https://iam-test.indigo-datacloud.eu/] : https://iam-tl-computing.cloud.cnaf.infn.it/  
Space delimited list of scopes or 'max' [openid profile offline_access]:  
Registering Client ...  
Generating account configuration ...  
accepted  
To continue and approve the registered client visit the following URL in a Browser of your choice:  
https://iam-tl-computing.cloud.cnaf.infn.it/authorize?response_type=code&client_id=d70f9284-c371-4ce2-90e9-1cd3d6201329&redirect_uri=http://localhost:13481&scope=openid profile  
offline_access access_type=offline&prompt=consent&state=0:2s4EBiNy_Z0a835JBiaW_cGG:33:  
L3RtcC9vaWRjLTRzMjdBOS9vaWRjLWFNZNZw50LTg5YjgtZjI2NjFmMGZiZmE3IiwieyY2xpZW50X2lkIjoizDcwZjk5ODQtYzZMMS00Y2UyLTkwZTkxMTkxMDQ2MjJmZGZiZmE3In0.irmwiCxdHZZWOpv1l9jbmgExev1_gyb9fhHENLSjbfOL4WV--vrh3E3pLu-elFV8c23TexwbHvnJGEZXaoQvNJMO3VUaso2ioZIZ2oPTTEmxJpa52goNTYrBrm2TRGERKAE0VaEqJbbI3wDoCJ_BPkdDLAT0YL5-AbHTkum8Ovsim-Dbnj_Bf3BpzEOt4  
/bin/xdg-open: line 881: www-browser: command not found  
/bin/xdg-open: line 881: links2: command not found  
/bin/xdg-open: line 881: elinks: command not found  
/bin/xdg-open: line 881: links: command not found  
Polling oidc-agent to get the generated account configuration ...success  
The generated account config was successfully added to oidc-agent. You don't have to run oidc-add.  
Enter encryption password for account configuration 'fornariclas12':  
Confirm encryption Password:  
  
[fornariclas12@ui-tier1 ~]$ oidc-token fornariclas12  
eyJraWQiOiJyc2ExIiwiaWxnImJoUlMyNTYifQ.  
eyJzdWIiOiJkbGhtNTQ5OC01ODlhLTQyYyItODJhMi04M2U2MjY3NTQwMQiLCJpc3MiOiJodHRwc2pcLlwwaWFTLXQxLWNvbXBldGlucy5jbG91ZC5jbmcFmlmLm4uaXRclLyIsIm5hbWUiOiJGZWRLcmllb2YBGb3JuYXJpIiwidmVhdXBzIjpibWImZhemhlhiIiwiaWxzYyEyIl0sInByZWZlcncnJlZF91ZS2YbmFtZSI6ImZmb3JuYXJpIiwib3N0YW5pc2F0aW9uX25hbWUiOiJOMSljb2lwZXRpbmciLCJleHAiOiJlMjMjA5MDYxNDksImhhCI6MTYyMDkwMjU0OSwianRpIjoijm0Y2VjnzMtMDAwMC0MGU0LTg5YjgtZjI2NjFmMGZiZmE3IiwieyY2xpZW50X2lkIjoizDcwZjk5ODQtYzZMMS00Y2UyLTkwZTkxMTkxMDQ2MjJmZGZiZmE3In0.irwbiCxdHZWOpyv1l9jbmgExev1_gyb9fhHENLSjbfOL4WV--vrh3E3pLu-elFV8c23TexwbHvnJGEZXaoQvNJMO3VUaso2ioZIZ2oPTTEmxJpa52goNTYrBrm2TRGERKAE0VaEqJbbI3wDoCJ_BPkdDLAT0YL5-AbHTkum8Ovs
```

```

gfal-rm https://xfer-archive.cr.cnaf.infn.it:8443/jlab12/test-fornari

[fornariclas12@ui-tier1 ~]$ condor_submit -spool -name sn-02 test.sub
Submitting job(s).
1 job(s) submitted to cluster 15750247.

[fornariclas12@ui-tier1 ~]$ watch condor_q -name sn-02 15750247

[fornariclas12@ui-tier1 ~]$ condor_transfer_data -name sn-02 15750247
Fetching data files...

[fornariclas12@ui-tier1 ~]$ cat outfile.txt
this is a test by F. Fornari
test-fornari
total 512
-rw-r-----+ 1 storm storm 29 May 13 15:29 test-fornari
[fornariclas12@ui-tier1 ~]$ cat errors.txt
[fornariclas12@ui-tier1 ~]$ cat test.log
000 (15750247.000.000) 05/13 15:28:44 Job submitted from host: <131.154.192.58:9618?addrs=131.154.192.58-9618&noUDP&sock=703648_945d_3>
...
001 (15750247.000.000) 05/13 15:29:29 Job executing on host: <131.154.208.195:9618?addrs=131.154.208.195-9618&noUDP&sock=14671_aa08_3>
...
006 (15750247.000.000) 05/13 15:29:30 Image size of job updated: 237380
    0 - MemoryUsage of job (MB)
    0 - ResidentSetSize of job (KB)
...
005 (15750247.000.000) 05/13 15:29:30 Job terminated.
    (1) Normal termination (return value 0)
        Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
        Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
        Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
        Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
    136 - Run Bytes Sent By Job
    544 - Run Bytes Received By Job
    136 - Total Bytes Sent By Job
    544 - Total Bytes Received By Job
    Partitionable Resources :      Usage  Request  Allocated
        Cpus                :      0.02      1          1
        Disk (KB)           :      25      1    3400414
        Memory (MB)         :      0      1      128
...

[fornariclas12@ui-tier1 ~]$ oidc-agent-service stop
unset OIDC_SOCKET;
unset OIDCD_PID;
echo Agent pid 24806 killed;
unset OIDCD_PID_FILE;

```