



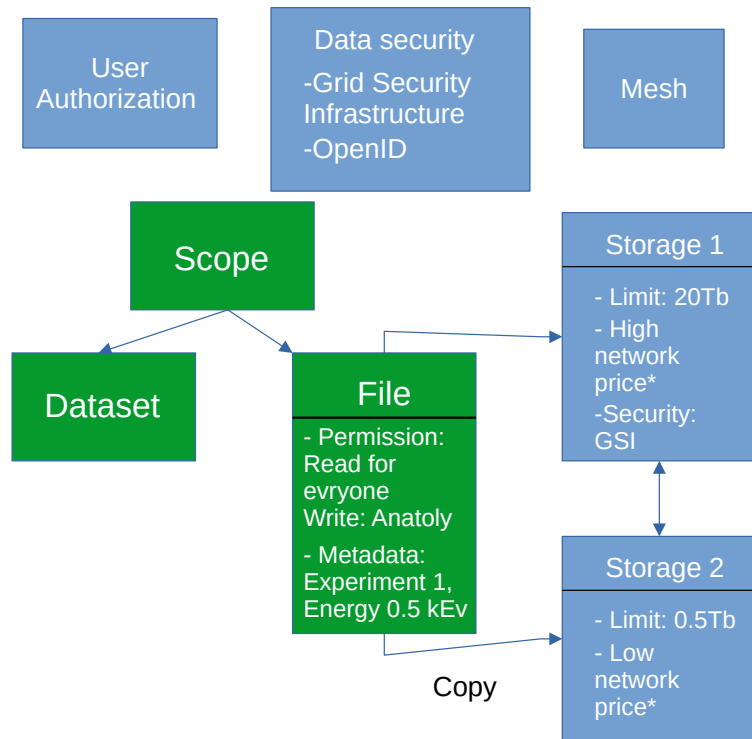
# Integration of modern distributed big data storage management systems for the BM@N particle collision experiment of the NICA project

Anatoly Aleksandrov



# Introduction

## Data Management System



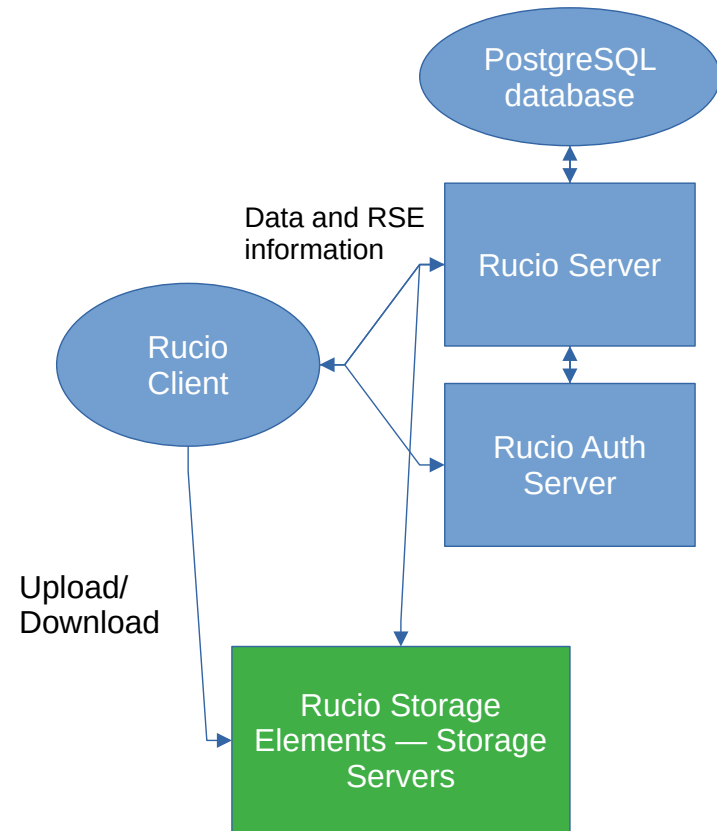
Modern data management systems have more functionality than in the diagram

# Software stack

- **Rucio as data management system**
- **GFAL2 compatible storage system: XROOTD, GridFTP, SRM, etc.**
- **PostgreSQL — RDBMS for Rucio database**

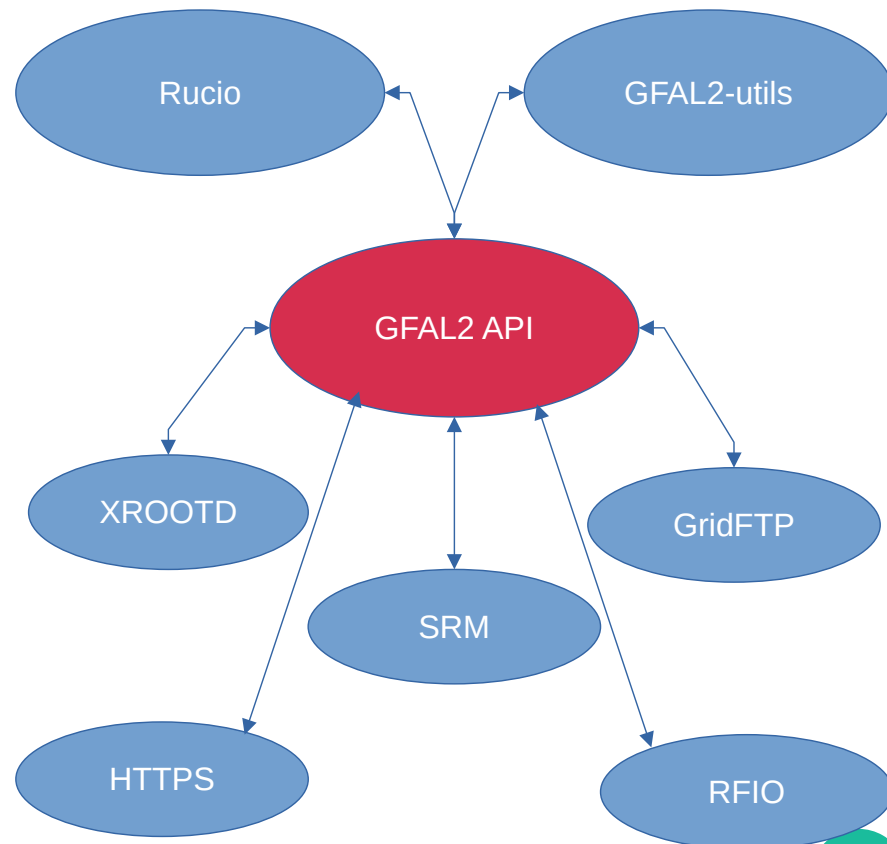
# Rucio data management system

- Server-client application
- Support GFAL2 interface
- Built-in user authorization system (X509, OpenID)



# GFAL2 — multi-protocol data management library

One interface combining data protocol interfaces used in the ATLAS project and other CERN projects



# Deployment automation

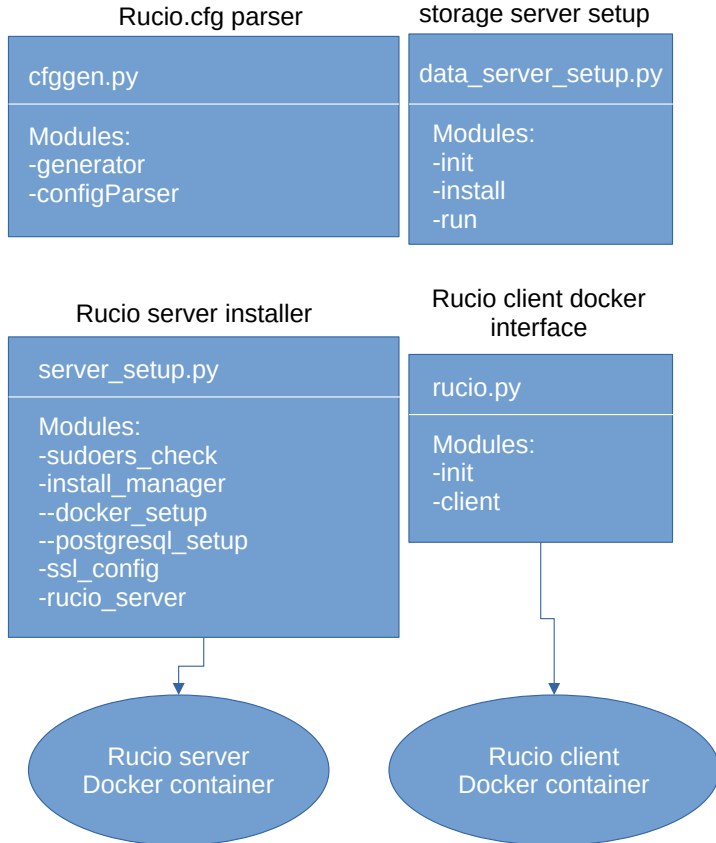
## To deploy the Rucio server:

- 1) Create SSL certificate with Certification Authority certificate (Rucio not working without SSL)
- 2) Install and configure PostgreSQL (or another RDBMS)
- 3) Create Rucio database and initialize it, add admin username and password
- 4) Install and configure Rucio Server

## If you use my python script:

- 1) Run my python script.
- 2) Enter admin username and password.
- 3) Enjoy your Rucio server!

# Scripts



- `server_setup.py` — rucio server setup script
- `storage_setup.py` — storage server setup script
- `rucio.py` — rucio client
- `cfggen.py` — command line interface for writing rucio client «Rucio.cfg» files

# server\_setup.py — rucio server script

server\_setup.py

Modules:

- sudoers\_check
- install\_manager
- docker\_setup
- postgresql\_setup
- ssl\_config
- rucio\_server

Usage manual:

```
[anatoly@archlinux server]$ sudo python3 server_setup.py --help
```

Usage:

```
python3 server_setup.py --init postgresql --init rucio --ssl --password mypassword --domain mydomain.org
```

args:

"--init postgresql" for install and initialize database (use with --password arg for set db password)

"--init rucio" initialize rucio server (u need init database)

"--ssl" for secure (use with --domain arg for set domain name)

"--domain mydomain.org" set domain for ssl certificate

"--password mypassword" set PostgreSQL password for init database



# Accounts

```
[anatoly@archlinux rucio]$ python3 rucio.py admin account list
account list
root
```

```
[anatoly@archlinux rucio]$ python3 rucio.py admin account info root
account info root
suspended_at : None
account      : root
account_type : SERVICE
email       : None
created_at  : 2024-04-28T16:59:41
status      : ACTIVE
deleted_at  : None
updated_at  : 2024-04-28T16:59:41
```

```
[anatoly@archlinux rucio]$ python3 rucio.py admin account get-limits root
account get-limits root
usage: rucio-admin account get-limits
       [-h]
       account
       rse
       [{local,global}]
rucio-admin account get-limits: error: the following arguments are required: rse
```

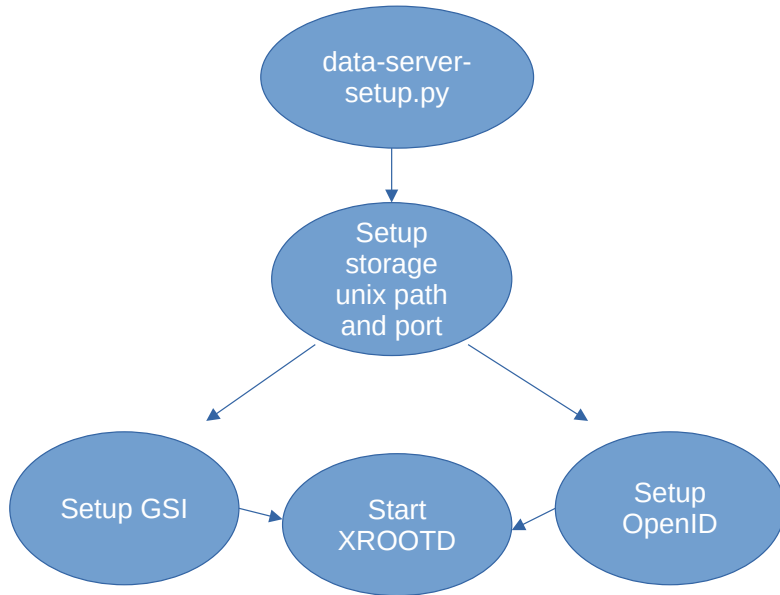
```
[anatoly@archlinux rucio]$ python3 rucio.py admin account get-limits root TEST_RSE
account get-limits root TEST_RSE
Quota on TEST_RSE for root : Inf
```

- Accounts:
- File permissions
  - Storage limits

# Rucio Storage Elements

```
Attributes:
=====
  TEST_RSE: True
Protocols:
=====
  root
    domains: '{"lan": {"read": 1, "write": 1, "delete": 1}, "wan": {"read": 1, "write": 1, "delete": 1, "third_party_copy_read": 1, "third_party_copy_write": 1}}'
    extended_attributes: None
    hostname: 192.168.122.192
    impl: rucio.rse.protocols.gfal.Default
    port: 1094
    prefix: //data
    scheme: root
Usage:
=====
  rucio
    files: 0
    free: None
    rse: TEST_RSE
    rse_id: c0a8aa2123ce43f3853e55b6657502ec
    source: rucio
    total: 0
    updated_at: 2024-04-28 17:56:34
    used: 0
RSE limits:
=====
```

# XROOTD protocol



## Secure libs:

- LibXrdSec.so
- libXrdSecgsi.so

XROOTD config file:

```
anatoly@anatoly:~$ cat xrootd.cfg
xrootd.export /data
xrootd.chksum md5
xrootd.seclib /usr/lib/x86_64-linux-gnu/libXrdSec.so
sec.level all standart
sec.protocol gsi -crl:3 -cert:~/rucio_certs/ruciocrt.pem -gridmap:~/grid-mapfile -key:~/rucio_certs/ruciokey.pem -certdir:~/rucio_certs
```

# Main Rucio client commands

```
[anatoly@archlinux rucio]$ python3 rucio.py upload --scope test --rse TEST_RSE testfile2.txt
```

```
2024-05-14 11:02:58,851 INFO      Preparing upload for file testfile2.txt
2024-05-14 11:02:58,891 INFO      File DID already exists
2024-05-14 11:02:59,221 INFO      Trying upload with root to TEST_RSE
2024-05-14 11:02:59,329 INFO      Successful upload of temporary file. root://192.168.122.192:1094//data/test/29/46/testfile2.txt.rucio.upload
2024-05-14 11:04:02,458 INFO      Successfully uploaded file testfile2.txt
```

```
[anatoly@archlinux rucio]$ python3 rucio.py download test:testfile2.txt
```

```
2024-05-14 11:06:51,731 INFO      Processing 1 item(s) for input
2024-05-14 11:06:52,282 INFO      No preferred protocol impl in rucio.cfg: No section: 'download'
2024-05-14 11:06:52,283 INFO      Using main thread to download 1 file(s)
2024-05-14 11:06:52,283 INFO      Preparing download of test:testfile2.txt
2024-05-14 11:06:52,331 INFO      Trying to download with root and timeout of 360s from TEST_RSE: test:testfile2.txt
2024-05-14 11:06:52,403 INFO      Using PFN: root://192.168.122.192:1094//data/test/29/46/testfile2.txt
/usr/local/lib/python3.9/site-packages/urllib3/connectionpool.py:1061: InsecureRequestWarning: Unverified HTTPS requ
dthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
  warnings.warn(
2024-05-14 11:06:52,518 INFO      File test:testfile2.txt successfully downloaded in 0.01 seconds
```

```
-----
Download summary
-----
```

```
DID test:testfile2.txt
Total files (DID):           1
Total files (filtered):      1
Downloaded files:           1
Files already found locally: 0
Files that cannot be downloaded: 0
```

MD5 or Adler32  
functions strongly  
needed on data server

SCOPE:FILENAME

# Main Rucio client commands

## Metadata:

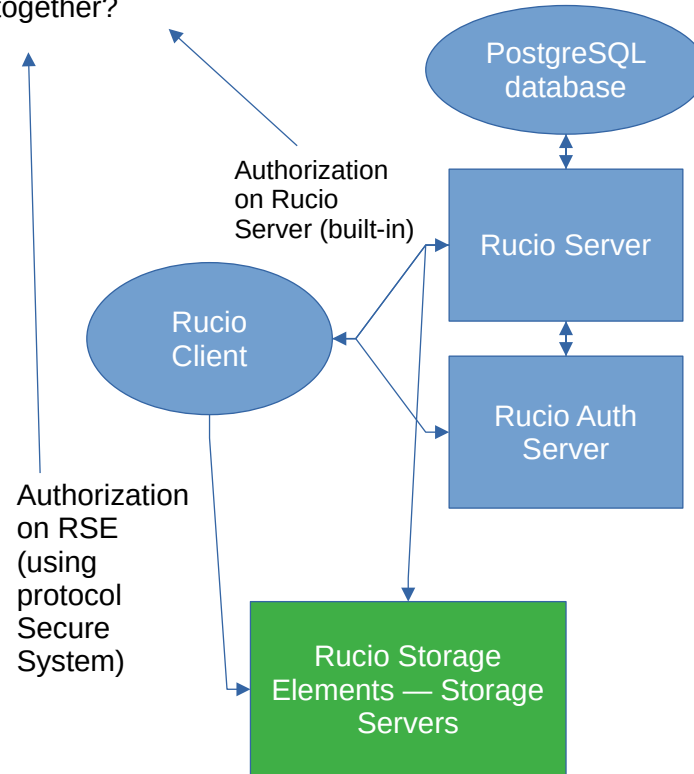
```
[anatoly@archlinux rucio]$ python3 rucio.py set-metadata --did test:testfile2.txt --key "Energy" --value 0.5-keV
```

```
[anatoly@archlinux rucio]$ python3 rucio.py get-metadata test:testfile2.txt --plugin ALL
```

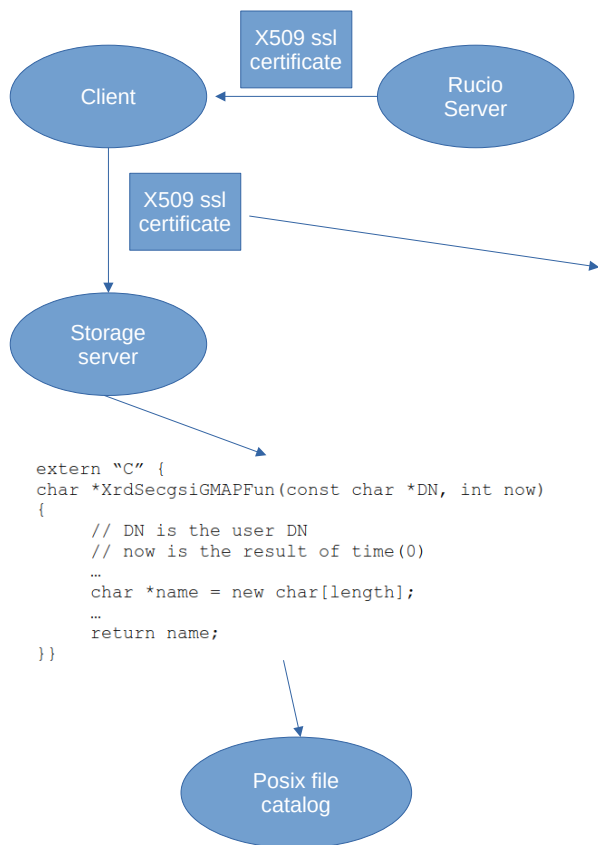
```
Energy: 0.5-keV
access_cnt: None
accessed_at: None
account: root
adler32: 00000001
availability: AVAILABLE
bytes: 0
campaign: None
closed_at: None
complete: None
constituent: None
created_at: 2024-05-14 11:01:29
datatype: None
deleted_at: None
did_type: FILE
```

# Authorization on data servers

How to make Rucio authorization system and third-party data protocols work together?



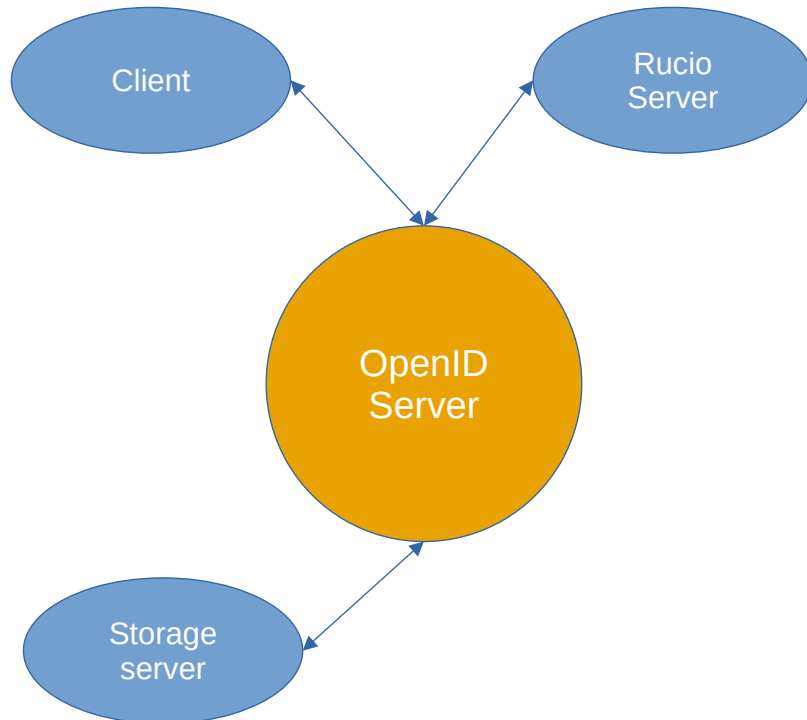
# Grid Security Infrastructure



<b>Subject Name</b>	
Country	US
State/Province	California
Organization	Stanford University
Common Name	*.slac.stanford.edu
<b>Issuer Name</b>	
Country	US
State/Province	MI
Locality	Ann Arbor
Organization	Internet2
Organizational Unit	InCommon
Common Name	<a href="#">InCommon RSA Server CA</a>
<b>Validity</b>	
Not Before	Wed, 19 Jul 2023 00:00:00 GMT
Not After	Thu, 18 Jul 2024 23:59:59 GMT
<b>Subject Alt Names</b>	
DNS Name	*.slac.stanford.edu
DNS Name	slac.stanford.edu

# OpendID & Rucio

Centralized  
authorization system



XROOTD have ztn  
protocol for work with  
tokens





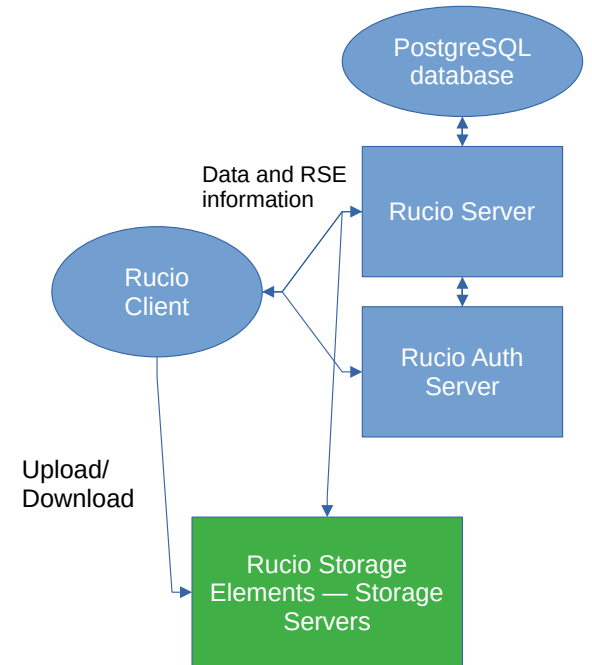
**Need more data protocols?**

# Adding your own protocols

## RSEProtocol

```
+ __init__(self, protocol_attr,  
rse_settings, logger=logging.log)  
  
+fns2pfns(self, lfns)  
  
+parse_pfns(self, pfns)  
  
+exists(self, path)  
  
+connect(self)  
  
+close(self):  
  
+get(self, path, dest,  
transfer_timeout=None)  
  
+put(self, source, target, source_dir,  
transfer_timeout=None)  
  
+delete(self, path)  
  
+rename(self, path, new_path)  
  
+get_space_usage(self)  
  
+stat(self, path)
```

Server and Client  
use this class



# Github project

- **Deployment scripts (in progress):**  
<https://github.com/proton-group/autorucio>
- **Rucio:** <https://github.com/rucio/rucio>
- **XROOTD:** <https://github.com/xrootd/xrootd>

**Thanks for your attention!**

## Command-Line Interface for work with GFAL2 data storages

- **gfal-copy**
- **gfal-ls**
- **gfal-mkdir**
- **gfal-rm**
- **etc**

# TestBench

```
ubuntu22.04 на QEMU/KVM
Файл  Виртуальная машина  Вид  Отправить комбинацию клавиш
[Icons] [Play] [Pause] [Power] [Refresh]

anatoly@anatoly:~$ ls
server
anatoly@anatoly:~$ cd server
anatoly@anatoly:~/server$ ls
ca  ctygen.py  docker-install.sh  PROXIES  postgres-install.sh  rucio_certs  rucio.cfg  server_setup.py
anatoly@anatoly:~/server$

Config warning: this hostname, anatoly, is registered without a domain qualification.
+++++ xrootd anon@anatoly initialization started.
Config using configuration file /home/anatoly/xrootd.cfg
Config maximum number of connections restricted to 1048576
Copy: 2012 Stanford University, xroot protocol 5.1.0 version v5.4.1
+++++ xroot protocol initialization started.
====> xrootd.export /data
====> xrootd.chksum md5
====> xrootd.seclib /usr/lib/x86_64-linux-gnu/libXrdSec.so
Config exporting /data
Plugin loaded secprot v5.4.1 from seclib /usr/lib/x86_64-linux-gnu/libXrdSec-5.so
+++++ Authentication system initialization started.
240514 09:52:55 1018 sec.Config: invalid level option - standart
====> sec.level all standart
Plugin loaded secgsi v5.4.1 from sec.protocol libXrdSecgsi-5.so
Secgsi -----
Secgsi Mode: server
Secgsi Debug: -1
Secgsi Ca dir: /rucio_certs
Secgsi Ca verification level: verifyfuss
Secgsi CRL dir: /etc/grid-security/certificates/
Secgsi CRL extension: .r0
Secgsi CRL check level: require
Secgsi CRL refresh time: 86400
Secgsi Certificate: /rucio_certs/rucio.crt.pem
Secgsi Key: /rucio_certs/rucio.key.pem
Secgsi Proxy delegation option: ignore
Secgsi GRIDmap file: /grid-mapfile
Secgsi GRIDmap option: trymap
Secgsi GRIDmap cache entries expiration (secs): 600
Secgsi VOMS option: ignore
Secgsi MonInfo option: 0
Secgsi Crypto modules: ssl
Secgsi Ciphers: aes-128-cbc:bf-cbc:des-ede3-cbc
Secgsi MDigests: sha1:md5
Secgsi Untrusting DNS for hostname checking
Secgsi -----
240514 09:52:55 1018 secgsi.GetSvcCertEnt: failed to load certificate from files (/home/anatoly/rucio_certs/rucio.crt.pem)
240514 09:52:55 1018 secgsi.Init: Secgsi: ErrInit: unable to generate ca cert hash list!
Secgsi: ErrInit: unable to generate ca cert hash list!
Config failed to load gsi authentication protocol!
====> sec.protocol gsi -cr193 -cert:/rucio_certs/rucio.crt.pem -gridmap:/grid-mapfile -key:/rucio_certs/rucio.key.pem
Config 2 authentication directives processed in /home/anatoly/xrootd.cfg
----- Authentication system initialization failed.
Config Unable to create security framework via /usr/lib/x86_64-linux-gnu/libXrdSec.so
240514 09:52:55 1018 xrootd.config: unable to load security system.
----- xroot protocol initialization failed.
240514 09:52:55 1018 XrdProtocol: Protocol xroot could not be loaded
----- xrootd anon@anatoly-1 initialization failed.
anatoly@anatoly:~$
```

- Demo environment based on 2 ubuntu virtual machine and host as client

# Rucio command-line interface

```
[anatoly@archlinux rucio]$ python3 rucio.py
```

```
usage: rucio
       [-h]
       [--version]
       [--config CONFIG]
       [--verbose]
       [-H ADDRESS]
       [--auth-host ADDRESS]
       [-a ACCOUNT]
       [-S AUTH_STRATEGY]
       [-T TIMEOUT]
       [--robot]
       [--user-agent USER_AGENT]
       [--vo VO]
       [-u USERNAME]
       [-pwd PASSWORD]
       [--oidc-user OIDC_USERNAME]
       [--oidc-password OIDC_PASSWORD]
       [--oidc-scope OIDC_SCOPE]
       [--oidc-audience OIDC_AUDIENCE]
       [--oidc-auto]
       [--oidc-polling]
       [--oidc-refresh-lifetime OIDC_REFRESH_LIFETIME]
       [--oidc-issuer OIDC_ISSUER]
       [--certificate CERTIFICATE]
       [--ca-certificate CA_CERTIFICATE]
       {ping,whoami,list-file-replicas,list-dataset-replic
d-container,attach,detach,ls,list-dids,list-dids-extended,
list-parent-datasets,list-scopes,close,reopen,stat,erase,l
ntent,list-content-history,upload,get,download,get-metadat
lete-metadata,list-rse-usage,list-account-usage,list-accou
```

{ping,whoami,list-file-replicas,list-dataset-replicas,add-dataset,add-container,attach,detach,ls,list-dids,list-dids-extended,list-parent-dids,list-parent-datasets,list-scopes,close,reopen,stat,erase,list-files,list-content,list-content-history,upload,get,download,get-metadata,set-metadata,delete-metadata,list-rse-usage,list-account-usage,list-account-limits,add-rule,delete-rule,rule-info,list-rules,list-rules-history,update-rule,move-rule,list-rses,list-suspicious-replicas,list-rse-attributes,list-datasets-rse,test-server,touch,add-lifetime-exception}

# Rucio admin cmd-line interface

```
[anatoly@archlinux rucio]$ python3 rucio.py admin
```

```
[sudo] пароль для anatoly:
```

```
usage: rucio-admin
```

```
[-h]
[--version]
[--verbose]
[-H ADDRESS]
[--auth-host ADDRESS]
[-a ACCOUNT]
[-S AUTH_STRATEGY]
[-T TIMEOUT]
[--vo VO]
[-u USERNAME]
[-pwd PASSWORD]
[--oidc-user OIDC_USERNAME]
[--oidc-password OIDC_PASSWORD]
[--oidc-scope OIDC_SCOPE]
[--oidc-audience OIDC_AUDIENCE]
[--oidc-auto]
[--oidc-polling]
[--oidc-refresh-lifetime OIDC_REFRESH_LIFETIME]
[--oidc-issuer OIDC_ISSUER]
[--certificate CERTIFICATE]
[--ca-certificate CA_CERTIFICATE]
{data,account,identity,rse,scope,config,subscription,replicas}
```

**Rucio admin client allows you to configure and administer the system**

# In progress

- **Authorization**
- **Protocols**
- **Scripting**