DSS documentation for data curators

1. Introducing DSSWeb
   1.1. Introducing the DSSWeb Management Portal
   1.2. Introducing the DSSWeb Management CLI
   1.3. Introducing the DSSWeb Management API

2. Understanding Data Projects and Data Pools
   2.1. Getting information about the data projects of a data curator
   2.2. Getting information about the grants for a particular data project

3. Understanding Containers
   3.1. Understanding Data Science Storage Containers

4. Understanding the Asynchronous Task Model
   4.1. Understanding Task Failures

5. Understanding our Software Development Process

6. Managing Containers
   6.1. Creating Containers
   6.2. Showing Containers
   6.3. Modifying Containers
   6.4. Delete Containers

7. Managing Container Access Rights
   7.1. Granting Access Rights
   7.2. Showing Access Rights
   7.3. Modify Access Rights
   7.4. Revoking Access Rights

8. Managing DSS Container Access Rights for External Users via Globus
   8.1. Granting Globus Sharing Access Rights
   8.2. Showing Globus Sharing Access Rights
   8.3. Revoking Globus Sharing Access Rights

9. Managing DSS Container Auto Group Links
   9.1. Creating Auto Group Links
   9.2. Showing Auto Group Links
   9.3. Updating Auto Group Links
   9.4. Deleting Auto Group Links

10. Managing DSS Container Exports
    10.1. Limitations
    10.2. Creating exports
    10.3. Showing exports
    10.4. Deleting exports

11. Hints and possible pitfalls
    11.1. Known Limitations
    11.2. Do's and Don'ts

1. Introducing DSSWeb

As a data curator of one or more data projects, you can manage your DSS storage via our DSSWeb self-service system. The system provides you with a Web Interface (GUI), a Command Line Interface (CLI) and a HTTP REST API (API).

You can think of DSSWeb as your virtual storage system administrator, which accepts your request, such as:

- Create a 10TB DSS container for project XYZ and back it up every day
- Give user Max access to DSS container XYZ
- Export DSS Container XYZ to my virtual machine with IP X.X.X.X for 10 days

and then automatically executes the necessary steps in order to fulfill your request.

1.1. Introducing the DSSWeb Management Portal

The DSSWeb Management User Interface is accessible via https://dssweb.dss.lrz.de. Your user needs to have data curator rights of a least one data project, to log in.

1.2. Introducing the DSSWeb Management CLI

The DSSWeb Command Line Interface is currently available as Software as a Service (SaaS) in the LRZ cloud (which we recommend to use) and as an on-premise software solution. In the following, we will describe how to get started with both versions of the CLI.

1.2.1. Installing the CLI on premise

In order to install the CLI on premise, the installation of git and python3.4 (or higher) is required. Use the following steps to install the CLI on a *NIX system:
git clone https://gitlab.lrz.de/a2822bp/dssweb-client.git
cd dssclient
python3 setup.py install

After successful installation you should be able to start the CLI in interactive mode using the dsscli command.

$ dsscli
(dsscli) help

Shell commands (type help <topic>):
====================================
cmdenvironment help load pyscript run set shortcuts
edit history py quit save shell show

Application commands (type help <topic>):
=========================================
complete dss container update login project show
dss container create dss nfsexport create pool list
dss container list dss nfsexport list pool show
dss container show dss nfsexport show id pool show id
dss container show id help project list
...

You can also use the CLI in non-interactive mode, like a regular shell command by calling the dsscli command directly with the particular arguments for the action you want to perform.

$ dsscli -h

DSSWeb Client

optional arguments:
  --version     show program's version number and exit
  -v, --verbose Increase verbosity of output. Can be repeated.
  -q, --quiet   Suppress output except warnings and errors.
  --log-file LOG_FILE Specify a file to log output. Disabled by
default.
  -h, --help    Show help message and exit.
  --debug       Show tracebacks on errors.

Commands:
...
  help           print detailed help for another command (cliff)
  login          Request username and password and obtain a token from
DSSWeb API.
  ...

1.2.2. Updating the CLI

As new features become available, we will update the code in the gitlab repository, so make sure to update the software from time to time. In order to update the software, just follow the installation procedure again. It will overwrite the old version of the software.
1.2.3. Accessing the CLI in the cloud

In order to access the interactive CLI, which is offered as Software as a Service in the LRZ cloud, just use SSH to login to dsscli.dss.lrz.de. In order to be able to log in, your user will need to have data curator rights for one or more data projects.

```
$ ssh XXXXXX@dsscli.dss.lrz.de
XXXXXX@dsscli.dss.lrz.de's password:
```

Shell commands (type help <topic>):
```
===================================
cmdenvironment help load pyscript run set shortcuts
edit history py quit save shell show
```

Application commands (type help <topic>):
```
=========================================
complete               dss container update login project show
dss container create dss nfsexport create pool list
dss container list dss nfsexport list pool show
dss container show dss nfsexport show id pool show id
dss container show id help project list
```

For your convenience, you can provide one or more SSH public keys in order to be able to login in without having to provide a password. Just use the edit command, which will allow you to edit the authorized_keys file of your user.

Another advantage of the SaaS version is that we will provide you with updated versions of the CLI automatically.

1.2.4. General usage of the CLI

The CLI is structured in several subcommands, that allow you to list, show, create and change various items of a data project like containers, access invitations, NFS exports, etc. To get a list of available subcommands, just call the help command in interactive mode or use the -h switch in non-interactive mode.

In order to get help on a particular command, you can just call the subcommand, followed by the -h switch.
List all available DSS data containers.

positional arguments:
  projectname

optional arguments:
  -h, --help            show this help message and exit
  --poolname [POOLNAME]

output formatters:
  output formatter options
  -f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}
      the output format, defaults to table
  -c COLUMN, --column COLUMN
      specify the column(s) to include, can be repeated

  table formatter:
    --max-width <integer>
      Maximum display width, <1 to disable. You can also use
      the CLIFF_MAX_TERM_WIDTH environment variable, but the
      parameter takes precedence.
    --fit-width
      Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable
    --print-empty
      Print empty table if there is no data to show.

  json formatter:
    --noindent
      whether to disable indenting the JSON

  CSV Formatter:
    --quote {all, minimal, none, nonnumeric}
      when to include quotes, defaults to nonnumeric

Note that the output of the commands is usually optimised for human readability. However, if you want to script certain things, you might be interested using the --f switch to change the output formatting to something, which is easier to parse by a script.

1.3. Introducing the DSSWeb Management API

Currently the API still lacks some important documentation information. Though, we are working hard on extending this documentation in the future. You can access the current documentation via https://dssweb.dss.lrz.de/docs/.
2. Understanding Data Projects and Data Pools

In the LRZ Data Science Management Concept, we use Data Projects as organizational envelopes around the data, which is stored in the context of this data project. This is particularly useful as it implements a link between the data and the people who are formally responsible for the data and therefore protects us from orphaned data.

A particular data project is managed by a group of users, which is called the *data curator group* of the project. By default only the *master users* of the project belong to this group, but upon request via the Servicedesk, LRZ can also add arbitrary other users to the list of data curators for a data project.

In order to be able to store data in the context of a data project, the project also needs *grants* on one or more *data pools*. Data pools are physical storage systems which are integrated into the LRZ Data Science Management Ecosystem. Currently only Data Science Storage systems are supported, but it is planned to add other systems like an Archive for example in the future. Grants specify which fraction of a particular data pool (a quota), a particular data project is allowed to use. These grants are managed by LRZ.

Note that data pools are usually optimized for a particular data pattern or workload. This information is provided you via the description and hints fields of the data pool to give you some guidance on container placement.

2.1. Getting information about the data projects of a data curator

Click here to see how it works in the GUI

2.1.1. Using the GUI

When you log in to the DSSWeb GUI, you will be automatically redirected to the *Project Overview* site, which will give you all necessary information about the data project(s) you are curator of. In order to drill down to other information, you'll first have to select the project you want to work with. You can always come back to the Projects site and select a different project, if you need to switch context.
2.1.2. Using the CLI

In order to list the available data projects, for which you are assigned the data curator role, use the `project list` command.

```
(dsscli) project list
+--------+--------+
| ID     | Name   |
|--------+--------+
| 1      | pr53da |
| 2      | pr53ve |
| 3      | pr74cu |
| 4      | pr74qo |
| 5      | pr74wa |
+--------+--------+
```

In order to get the details about a particular data project, use the `project show` command.
2.2. Getting information about the grants for a particular data project

Click here to see how it works in the GUI

2.2.1. Using the GUI

Once you've selected a project to work with, you'll be automatically redirected to the Project Detail site.

Now click on the DSS Pools Icon and you will be redirected to the DSS Pool Overview site, which will list you all available storage grants for your project.
The diagram will show you, how much of your storage grant has already been allocated to containers by you or your data curator colleagues.

Click here to see how it works in the CLI

2.2.2. Using the CLI

In order to list the available grants for a particular data project, use the `pool list` command.

```plaintext
(dsscli) pool list pr74cu
+--------------------------------+-------------------+-------------------------+
| ID   | Project | Name                          |
+--------------------------------+-------------------+-------------------------+
| 7    | pr74cu  | SFA12XX Large File Pool       |
| 8    | pr74cu  | DSS Testsystem                |
+--------------------------------+-------------------+-------------------------+
```

In order to get the details about a particular data project, use the `pool show` command.
3. Understanding Containers

Containers are the organizational elements, in which your actual data will live. Depending on the type of the underlying storage system, containers could be implemented in very different ways. For the Data Science Storage (DSS) systems, a container is basically a directory in...
a POSIX file system, with some additional magic attached to it. For a future archive system or an object storage system, a container will most likely be implemented in a different way. However, what is important to note is the basic concept of a container, which will be equal across all potential underlying storage systems:

1. A container is some kind of storage element with a certain size, which lets you store data in it.
2. A container should be used to encapsulate data, which semantically belongs together.
3. You can invite arbitrary users, which exist in the LRZ Identity Management System to access the data in the container.

### 3.1. Understanding Data Science Storage Containers

As already discussed, data containers on a Data Science Storage system are implemented as a directory on a POSIX file system, which is owned by a POSIX group, dedicated to the container. Each container has a certain limitation on the capacity and number of files, which can be stored in the container. The members of the container group are managed by the data curator via an invitation process. The access to the container is managed via POSIX rights and NFSv4 style ACLs. Also each container is associated with a so called enforcement policy. This policy defines how hard the system will try to make sure that access rights and ACLs stay in alignment with the access rights concept. Last but not least, each DSS type container is associated with a so called protection policy, which defines if and how the data in a container is protected by regular backups to the LRZ Backup- and Archive system.

#### 3.1.1. Understanding Data Science Storage Container Enforcement Policies

In contrast to other file system types, such as NTFS, you cannot revoke certain POSIX mode bit and ACL change rights of the owner of a file. Therefore you cannot guarantee, that a user may accidentally or on purpose do something unexpected, that may break the rights enforcement methods in place. However, there are certain measurements that can be taken, which increasingly deviate from the POSIX standard and also are increasingly expensive in terms of performance. Therefore, we offer several levels of enforcement policy, so you can pick the right one for your use-case.

- The most basic enforcement policy is called **NONE**. Using this policy, the container directory will be created with owning user `root` and owning group `<container group>` and sets the POSIX rights to `rwxrwxr-`. This will make sure that directories and files created within the container will automatically inherit the container group setting and that directories created within the container will inherit the `setgid` bit. However, every user can change the permissions of their files and directories using chmod and therefore overwrite the default settings easily. So the use case for this policy is usually that you want to let your users manage the access rights within the container by themselves and we only make sure that access at the container level is restricted to the invited container users.
- A more restrictive but still almost POSIX compliant enforcement policy is called **NORMAL**. This is also the default mode, you should choose if you are in doubt. This implements the same POSIX mode bits as the `NONE` mode, but in addition sets a set of NFSv4 style ACLs on the container directory, that will automatically be inherited by all files and directories, created in the container, which makes sure that all users belonging to the container group have `rw-` rights on files and `rwx` rights on directories. The point where this setting deviates from the POSIX or NFSv4 semantic is that if a user changes the POSIX mode bits using `chmod`, we do not replace the ACLs by the new mode bits, but instead only update the ACL and therefore keep our access right settings for the container group intact. So basically the only way for a user to overwrite the permissions on a file he owns, is by explicitly editing the NFSv4 style ACL.
- A even more restrictive and barely POSIX compliant enforcement policy is called **STRICT**. This implements the same techniques described for the `NORMAL` mode, but disallows the use of `chmod` completely. As this happens on the file system level, this also holds true for each program library call, that uses a `chown` POSIX API call. So users should not be able to remove the `setgid` bit.
  
  However, by editing the ACLs, they still can overwrite the permissions.

#### 3.1.2. Understanding Container Access Rights Management

For managing container access rights, we follow an invitation approach as you may already be familiar from cloud storage services like the LRZ Sync+Share service or Dropbox, Google Drive, etc. This means, for example, if data curator Alice wants to give user Bob access to the data container Cont-A, Alice will tell DSSWeb to invite Bobs username(s) to access container Cont-A. DSSWeb will then send an E-Mail, containing a unique invitation link to the E-Mail address assigned to Bobs username in the LRZ Identity Management System. Bob can then accept the invitation by clicking on the invitation link and accepting our data privacy and usage terms.

As this happens on the file system level, this also holds true for each program library call, that uses a `chown` POSIX API call. So users should not be able to remove the `setgid` bit.

#### 3.1.3. Understanding Container Access Revocation and User Deletion Actions

As you may have noticed, the semantics of Data Containers are designed as collaborative space, in which every user can access all data. So semantically the data is not owned by an individual, but the whole group. However, as we operate on a POSIX file system, the POSIX semantics dictate that for each file there has to be a single user who is the formal owner of the file. Technically this is implemented by storing the UIDs of the owning user for each file in the metadata of the file.

Now let's look at the case when a data curator wants to revoke access of a certain user to a container. Therefore, we remove the user from the container group and because of the permissions set on the container level, the user cannot access any data in the container anymore. However, there may now still be files in the container which are formally owned by the revoked user. While this is technically insignificant - since the user cannot access the data in the container anymore - it may be misleading and ambiguous to the remaining users. The same holds true if a user, invited to a container gets completely deleted. In this case, even resolving the UID to a user name will not work anymore and you will be presented with a cryptic number as owner of the files in your container.
In order to avoid these kinds of problems, whenever you revoke access for a user or a user gets deleted in the Identity Management System, we automatically start a batch job, that changes ownership of all files, which belong to the revoked or deleted user. Please note that this will not happen if the container enforcement policy is set to NONE. Currently the new owner is automatically determined from the list of data curators, whereby we choose that user, which is the first one in an alphabetical order. Note that this happens asynchronously so there may be a noticeable delay between revocation/deletion and ownership change.

We plan to extend this functionality in future releases, so that you may be able to specify a new owner, when you revoke a particular user, or put all data owned by that user to a tape archive or even delete all data owned by that user. If you have a particular use case, which you wish to see implemented, please open a ticket at the LRZ Servicedesk so that we can discuss it and file a user story in our product backlog.

3.1.4. Understanding Automatic Container Access Group Links

There may be situations where explicitly managing DSS Container Access Rights for each individual user may represent unnecessary overhead, because you want to give one or more particular groups automatically access to a container. So when a new member is added to the group, he should automatically be invited to access the container and when a member is removed from the group he should automatically be revoked access rights.

You can achieve this by defining so called Auto Group Links for a container. We currently support to link groups from DSS and TUMonline and plan to extend this to LinuxCluster and SuperMUC groups in the future. For more details on this, please check out our Knowledge Base article on this topic.

Please note that invitations, created from Automatic Container Access Group Links have very limited manual management capabilities. The only thing, a data curator can manually do is to resend an invitation mail for such an invitation.

3.1.5. Understanding Data Science Storage Container Protection Policies

In order to avoid these kinds of problems, whenever you revoke access for a user or a user gets deleted in the Identity Management System, we automatically start a batch job, that changes ownership of all files, which belong to the revoked or deleted user. Please note that this will not happen if the container enforcement policy is set to NONE. Currently the new owner is automatically determined from the list of data curators, whereby we choose that user, which is the first one in an alphabetical order. Note that this happens asynchronously so there may be a noticeable delay between revocation/deletion and ownership change.

We plan to extend this functionality in future releases, so that you may be able to specify a new owner, when you revoke a particular user, or put all data owned by that user to a tape archive or even delete all data owned by that user. If you have a particular use case, which you wish to see implemented, please open a ticket at the LRZ Servicedesk so that we can discuss it and file a user story in our product backlog.

Please be aware that because of the typical size of multiple PBs of a single Data Science Storage system, the Recovery Time Objective for a catastrophic system failure, in which the data of a complete DSS system has to be restored is typically targeted to be one month.

- The most simple data protection policy is called NONE. Containers which are assigned this protection policy are not protected at all. Use this policy only for data, which can easily be regenerated or recaptured.
- For automatic backups, which are performed once a week, use the policy BACKUP_WEEKLY. This will perform an incremental backup run of your container once a week (usually during the weekend). The retention policies of the backup are 180 days and a maximum of 3 versions per file. For more information about DSS backup retention policies, see DSS Understanding Data Science Storage Container Backup Retention.
- For automatic backups, which are performed once a day, use the policy BACKUP_DAILY. This will perform an incremental backup run of your container every day (usually during night). The retention policies of the backup are 180 days and a maximum of 3 versions per file. For more information about DSS backup retention policies, see DSS Understanding Data Science Storage Container Backup Retention.

For containers which exclusively store static, read-only content, like simulation results, output data from instruments like genomic sequencers, microscopes, NMR spectrosopes, etc, we also provide special ARCHIVE policies. These policies help you to be compliant with the DFG rules of good scientific practice, which require you to retain scientific primary data for 10 years. This is implemented by assigning the backed up data of your container a special retention policy, that keeps backups for 10 years and also keeps two copies of the backed up data in two different locations. Please note the restrictions that apply when using this data protection policy: DSS Understanding Data Science Storage Container Backup Retention.

- For automatic incremental archives of static data, which are performed once a week, use the policy ARCHIVE_WEEKLY. This will perform an incremental archive run of your container once a week (usually during weekend).
- For automatic incremental archives of static data, which are performed once a day, use the policy ARCHIVE_DAILY. This will perform an incremental archive run of your container once a day (usually during night).

Data Science Storage Container data protection mechanisms have several known limitations. Make sure that you fully understand them, before implementing a solution that relies on them.

For information on how to restore backed up files, please check this Knowledge Base article.
3.1.6. Understanding Data Science Storage Container Sharing with External Users

In addition to sharing container data with LRZ/TUM/LMU users, you can also share your container data with arbitrary external users, using the 3rd party Globus Sharing service. We have integrated Globus Sharing via their API into the DSSWeb Self-Service portal for your convenience, so you can easily manage Globus Sharing permissions via DSSWeb. However, in order to use Globus Sharing on a particular container, we have to do some special preparations, which are not necessary if you don't need Globus Sharing. Therefore, you have to explicitly enable Globus Sharing on your container. This can either be done at container creation time or anytime afterwards. You can also disable Globus Sharing at any time. After Globus Sharing has been enabled on your container, you can invite arbitrary external users to access your data via Globus by just specifying their Globus User ID or E-Mail address. They will then receive an invitation Mail from Globus that provides the necessary instructions in order to access the shared data.

For details on how Globus Sharing and the integration into DSS and DSSWeb works, please check this Knowledge Base article.

4. Understanding the Asynchronous Task Model

As we already described earlier, DSSWeb takes your high level requests (like create a new container) and automatically executes the necessary steps to fulfill your request. Every time you tell DSSWeb to create or change something, it must talk to one or more external systems, like LRZ SIM, LRZ’s Backup and Archive System, the underlying filesystems of DSS and so on and ask them to execute one or more tasks. So under the covers, DSSWeb is a fairly complex, highly distributed system which depends on multiple systems working seamlessly together. However, in such highly distributed systems, we have to accept that things may fail from time to time. Fortunately, these failures often have a temporary and very seldom a permanent nature. For example, think of the situation in which one of the subsystems is not available for whatever reason. This situation is usually fixed very fast and then everything works just fine again.

In order to hide all this complexity and not to bother you with temporary errors, thereby giving you the best possible user experience, we work with an asynchronous task model. That means, whenever you tell DSSWeb to create or change something, it will just store the necessary information in its database and make sure that an asynchronous task is started in the background. So while DSSWeb may still be busy, fulfilling your request, you can already go on and do other stuff and then check back later to see if your request was carried out successfully.

You can check the status of an object (like a container), by looking at its Status and Active Task fields. Typically when you first create an object it goes to a CREATE_PENDING state, meaning the object is currently being setup. When an object reaches a "steady state", meaning it was successfully created, its status is usually ACTIVE. When you change an object, it will go to a CHANGE_PENDING state and as soon as the changes have been carried out it will go back to ACTIVE again. The same semantics often are also true for deletion of objects, meaning the object will got to a DELETE_PENDING state and only after it has been successfully cleaned up on the subsystems, it is removed from the DSSWeb database.

In the GUI, the status is usually represented via Icons. You can get a textual representation of the state by simply put your mouse pointer on such a status icon.

Please note that currently, when an object is in transition, we do not accept any further changes to the object. So whenever an object is in a PENDING state, you have to wait until it is ACTIVE again to apply the next operation to this object.

4.1. Understanding Task Failures

Usually, when an asynchronous task hits a failure in a subsystem, the only thing you should notice is that the task may take longer than normal to execute (there is no special state indicating a problem for an object, it will simply stay in the PENDING state). The implemented semantics are that whenever a task encounters a failure, it will retry after some minutes until the task can finally be finished. DSSWeb monitors the execution times and number of retries of tasks and in case the retry count of a task gets unusually high, it will notify LRZ personal to take a look. (We currently also think about automatically raising an Incident at the LRZ Servicedesk for you, so that you get feedback on the status of the fixing procedures.)

5. Understanding our Software Development Process

Our software development process is loosely based on an agile approach and is highly driven by the demands of our customers. So, if you are missing a particular feature or would like to see something particular to be implemented in DSSWeb, please don’t hesitate and reach out to us by opening a request on the LRZ Servicedesk so we can discuss your requirements and record it in our product backlog.

6. Managing Containers

In the following we describe how you can create a new DSS Container for your data project.

6.1. Creating Containers

Please note that in order to create a new DSS container, the following requirements must be met:
You must be assigned the data curator role for the data project
Your data project must own a grant for the specific data pool in which you want to create the container
The size and file limits of the container must be less or equal to the available size and files in your pools grant

Please note that currently you cannot change the data protection mode or enforcement policy of a container. Since these changes may require substantially resources to perform a transition or may have unwanted side effects on your data, currently these properties can only be set at container creation time. So choose carefully at container creation time.

If you really require such a transition for an existing container, please contact us via the LRZ Servicedesk.

Click here to see how it works in the GUI

6.1.1. Using the GUI

Once you've selected a project to work with, you'll be automatically redirected to the Project Details site.

Now click on the DSS Containers icon and you will be redirected to the DSS Container Overview site, which will list you all containers for your project.
As long as you have storage grants for an additional container available, you can click on the *New DSS Container* icon, to add a new container.
The Fields which are marked with an asterix * are mandatory fields. The other fields are optional. Fill out the form and click on Submit to create your new container.

If all the container could be successfully created, the UI will show you a System Message, that indicates that the container has been created.

If there were any errors, you will receive a System Message that indicates which error occurred instead.

After container creation, you'll see the new container on the DSS Container Overview site, whereby the status icon will indicate that creation is still pending.
Once the creation process has finished, the status icon will automatically change to the active icon.

Click here to see how it works in the CLI

### 6.1.2. Using the CLI

In order to create a new container for a particular data project in a particular pool, use the `dss container create` command.

This command takes the following mandatory arguments:

- The name of the data project
- The name of the data pool
- The maximum allowed amount of data, which can be stored in the container in Gigabytes
- The maximum allowed number of files, which can be created in the container

Additionally, you can specify the following optional arguments:

- A description for the container.
- The data protection mode as described in the data protection section. (Default is NONE)
- The enforcement policy as described in the enforcement policy section. (Default is NORMAL)
- Whether Globus Sharing should be enabled for this container or not. (Default is OFF)

In the following example, we are going to create a new container

- for the data project pr74cu
- on the pool DSS Testsystem
- with a size of 1000GB
- and a maximal number of 10,000 files
- which will be backed up daily
- and has assigned a strict enforcement policy
- has Globus Sharing disabled

```
(dsscli) dss container create --projectname pr74cu --poolname 'DSS Testsystem' --size 1000 --files 10000 --dpmode BACKUP_DAILY --idmode STRICT --description 'DSS tutorial container' --globus-sharing OFF
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>57</td>
</tr>
<tr>
<td>Name</td>
<td>pr74cu-dss-0015</td>
</tr>
<tr>
<td>Description</td>
<td>DSS tutorial container</td>
</tr>
<tr>
<td>Status</td>
<td>CREATE PENDING</td>
</tr>
<tr>
<td>Active Task</td>
<td>8e5bb484-ca57-4761-a630-000c1807f1d7</td>
</tr>
<tr>
<td>Read/Write Group</td>
<td>pr74cu-dss-0015</td>
</tr>
<tr>
<td>Read Only Group</td>
<td>pr74cu-dss-0015-ro</td>
</tr>
<tr>
<td>Quota GB</td>
<td>1000</td>
</tr>
<tr>
<td>Quota Files</td>
<td>10000</td>
</tr>
<tr>
<td>ID Streamline Mode</td>
<td>STRICT</td>
</tr>
<tr>
<td>Data Protection Mode</td>
<td>BACKUP_DAILY</td>
</tr>
</tbody>
</table>

Please note that all the data curators of a data project will automatically get access to the containers, created for their project. If that is not what you want, you have to revoke access for them manually.

### 6.2. Showing Containers

Please note that even in order to just access container information via DSSWeb, you currently need to be data curator for the particular project, the container belongs to.

Click here to see how it works in the GUI

### 6.2.1. Using the GUI
Once you've selected a project to work with, you'll be automatically redirected to the *Project Detail* site.

Now click on the *DSS Containers* icon and you will be redirected to the *DSS Container Overview* site, which will list you all containers for your project.
In order to get more details on a particular container, just click on Select and you will be redirected to the DSS Container Details site.

The Used Space diagram will show you, how much of the granted capacity and files for the container has already been used. Please note that this is not a real time display and also it may not be 100% accurate.
6.2.2. Using the CLI

In order to get an overview about all containers you are allowed to manage via DSSWeb, you can use the `dss container list` command. If you are data curator for multiple projects, you can limit the containers to the ones of a particular project. Additionally you can also limit the listing to the containers of a particular pool.

```
(dsscli) dss container list
```

```
+----+---------+----------------+-----------------+----------------+
| ID | Project | Pool           | Name            | Status         |
+----+---------+----------------+-----------------+----------------+
| 46 | pr74cu  | DSS Testsystem | pr74cu-dss-0008 | ACTIVE         |
| 48 | pr74cu  | DSS Testsystem | pr74cu-dss-0009 | ACTIVE         |
| 49 | pr74cu  | DSS Testsystem | pr74cu-dss-0010 | ACTIVE         |
| 50 | pr74cu  | DSS Testsystem | pr74cu-dss-0011 | ACTIVE         |
| 51 | pr74cu  | DSS Testsystem | pr74cu-dss-0012 | ACTIVE         |
| 55 | pr98kx  | DSS Testsystem | pr74cu-dss-0013 | ACTIVE         |
| 56 | pr98kx  | SFA12KX Large  | pr74cu-dss-0014 | CHANGE PENDING |
| 57 | pr74cu  | DSS Testsystem | pr74cu-dss-0015 | CREATE PENDING |
| 58 | pr74cu  | DSS Testsystem | pr74cu-dss-0016 | CREATE PENDING |
```

In order to view the details of a specific container, you can use the `dss container show` or `dss container show id` command. The first one takes the container name, the later one the container id of the container to show as mandatory argument:

```
(dsscli) dss container show pr74cu-dss-0002
```

```
+-----------------------+-----------------------------------------+
| Field                 | Value                                   |
+-----------------------+-----------------------------------------+
| ID                    | 133                                     |
| Project               | pr74cu                                  |
| Pool                  | DSS Testpool                            |
| Name                  | pr74cu-dss-0002                         |
| Path                  | /dss/dsstestfs01/pr74cu/pr74cu-dss-0002 |
| Description           | Testcontainer                           |
| Status                | ACTIVE                                  |
| Active Task           | None                                    |
| Read/Write Group      | pr74cu-dss-0002                         |
| Read Only Group       | pr74cu-dss-0002-ro                      |
| Quota GB              | 2048                                    |
| Used GB               | 9                                       |
| Used GB Percentage    | 0.44                                    |
```
6.3. Modifying Containers

In the following we show how certain properties of containers can be changed. Currently you can change the following properties of an existing container:

- The maximal size of the container (Quota GB)
- The maximal number of files of the container (Quota Files)
- The description
- Weather Globus Sharing

Click here to see how it works in the GUI

6.3.1. Using the GUI

Once you've selected a project to work with, you'll be automatically redirected to the Project Detail site.
Now click on the *DSS Containers* icon and you will be redirected to the *DSS Container Overview* site, which will list you all containers for your project.
In order to change settings for a particular container, just click on the container's Edit Button and you will be redirected to the DSS Container Update site.

Fill out the form and click on Submit to update your container.

If the container could be successfully changed, the UI will take you to the DSS Container Details site and show you a System Message, that indicates that the container has been updated.
After container update, you'll see the new container on the DSS Container Overview site, whereby the status icon will indicate that update is still pending.

Once the creation process has finished, the status icon will automatically change to the ACTIVE Icon and the Edit Button will be activated again.

Please note that you can only apply new changes to a container if it is in the ACTIVE state. If the container is not in the ACTIVE state, the Edit Button will be automatically disabled.

Click here to see how it works in the CLI

6.3.2. Using the CLI

In order to modify an existing container, use the dss container update command.

This command takes the following mandatory arguments:

- The name of the data container

Additionally, you have to specify at least one of the following arguments:

- The new maximum usable size.
- The new maximum number of files.
- The new description
- Whether Globus Sharing should be enabled for this container or not.

In the following example, we are going to update the container pr74qo-dss-0002:

- New maximum size: 1,000 GB
- New maximum files: 1,000,000
- New description: Updated description

```
(dsscli) dss container update pr74qo-dss-0002 --size 2048 --files 200000 --description 'Testcontainer updated' --globus-sharing ON
```

+----------------------+--------------------------------------+
| Field                | Value                                |
+----------------------+--------------------------------------+
| ID                   | 142                                  |
### 6.4. Delete Containers

Please note that we have not yet implemented a function for automatically deleting a data container. This feature will be released in a future version of DSSWeb. In the meantime, if you require the deletion of a data container, please contact the LRZ Service Desk.

### 7. Managing Container Access Rights

As discussed, Data Science Containers are collaborative spaces. As such, you can grant virtually any user, which is known in the LRZ Identity Management, access to a data container. As the Identity Management Systems of LRZ, TUM, LMU are kind of federated, this also means that you can grant any user, managed by one of these IdMs access to your data containers.

Please note that the various ways in which users can access the data within a container is documented in the DSS documentation for users document.

### 7.1. Granting Access Rights

In order to allow an user to access a particular data container, you must invite him or her to the container group.

> Click here to see how it works in the GUI

#### 7.1.1. Using the GUI

In order to invite a user to access a particular data container, navigate to the Container Detail site.

On this detail page, you'll find the Container Users section:
Click on the **Add new user button** and you will be redirected to the **Invitation Create site**.

The Fields which are marked with an asterix * are mandatory fields. The other fields are optional. Fill out the form and click on Submit to create your new invitation.

If all the invitation could be successfully be created, the UI will take you back to the **DSS Container Details site** and show you a **System Message**, that indicates that the invitation has been created.

After invitation creation, you’ll see the new invitation on the **DSS Container Overview site**, whereby the status icon will indicate that creation is still pending.
Once the creation process has finished, the status icon will automatically change to the USER ACCEPT PENDING state and once the user has accepted the invitation it will automatically change to the USER ACCEPTED state.

Please note that you will not see first name and last name until the invitee has accepted the invitation.

Click here to see how it works in the CLI

7.1.2. Using the CLI

In order to invite a user to access a particular data container, use the `dss invitation create` command.

This command takes the following mandatory arguments:

- The name of the data container
- The username to invite

Additionally, you can specify the following optional arguments:

- The access mode to grant. This can either be **READ_WRITE** (the default) or **READ_ONLY**. However, the later can only be used with containers that implement an enforcement policy of **STRICT**.
- The maximum amount of GB, the user is allowed to consume within the container. (Default: unlimited)
- The maximum number of files, the user is allowed to create within the container. (Default: unlimited)

In the following example, we are going to invite user `a28ditix` to container `pr74cu-dss-0000`, using:

- Maximum granted size: 1,000 GB
- Maximum granted files: 1,000,000
- Access mode: **READ_WRITE**

```
(dsscli) dss invitation create --containername pr74qo-dss-0001 --username di57gix --accessmode READ_WRITE --maxgb 1000 --maxfiles 1000000
```

+-------------+--------------------------------------+
| Field       | Value                                |
+-------------+--------------------------------------+
| ID          | 83                                   |
| Container   | pr74qo-dss-0001                      |
| User        | di57gix                              |
| Quota GB    | 1000                                 |
| Quota Files | 1000000                              |
| Access Mode | RW                                   |
| Status      | CREATE PENDING                       |
| Active Task | 46d9dc61-7135-4523-baa4-d950ecce2ba3 |
| Expires At  | 2018-01-04                           |
| Inviter     | a2828yy                              |
+-------------+--------------------------------------+

Please note that you will not see first name and last name until the invitee has accepted the invitation.

Note that size and file quotas are only accepted for invitations with **READ_WRITE** access. If you specify quotas for an invitation with **READ_ONLY** access, these will be ignored.

In the following example, we are going to invite user `a28ditix` to container `pr74cu-dss-0000`, using:

- Maximum granted size: 1,000 GB
- Maximum granted files: 1,000,000
- Access mode: **READ_WRITE**

```
(dsscli) dss invitation create --containername pr74qo-dss-0001 --username di57gix --accessmode READ_WRITE --maxgb 1000 --maxfiles 1000000
```

+-------------+--------------------------------------+
| Field       | Value                                |
+-------------+--------------------------------------+
| ID          | 83                                   |
| Container   | pr74qo-dss-0001                      |
| User        | di57gix                              |
| Quota GB    | 1000                                 |
| Quota Files | 1000000                              |
| Access Mode | RW                                   |
| Status      | CREATE PENDING                       |
| Active Task | 46d9dc61-7135-4523-baa4-d950ecce2ba3 |
| Expires At  | 2018-01-04                           |
| Inviter     | a2828yy                              |
+-------------+--------------------------------------+
7.2. Showing Access Rights

7.2.1. Using the GUI

In order to display the access rights of a container user, please navigate to the Container Detail site.

On this detail page, you'll find the Container User section:

![Container User Section](image)

7.2.2. Using the CLI

In order to get an overview about all invitations to containers of your project via DSSWeb, you can use the `dss invitation list` command. Additionally you can limit the output to the invitations of a particular container and/or username.

In the following examples, we will examine invitations for the example project `pr74qo`.

```
(dsscli) dss invitation list pr74qo
+-----------------+---------+-------------+----------------------+
| ID     | Container | User      | Access Mode | Status                |
|--------+-----------+-----------|------------|-------------|-----------------------|
+--------+-----------+-----------+------------|-------------|-----------------------|
| 80     | pr74qo-dss-0000 | a2828xx | RW         | USER ACCEPTED         |
| 81     | pr74qo-dss-0000 | di57gix  | RW         | USER ACCEPTED         |
| 82     | pr74qo-dss-0001 | a2828xx | RW         | USER ACCEPTED         |
| 83     | pr74qo-dss-0001 | di57gix  | RW         | USER ACCEPT PENDING   |
+--------+-----------+-----------+------------|-------------|-----------------------+
In the following examples, we will examine invitations for the example container pr74qo-dss-0000.

```plaintext
(dsscli) dss invitation list --containername pr74qo-dss-0000
+----+-----------------+---------+-------------+---------------------+
<table>
<thead>
<tr>
<th>ID</th>
<th>Container</th>
<th>User</th>
<th>Access Mode</th>
<th>Status</th>
</tr>
</thead>
</table>
+----+-----------------+---------+-------------+---------------------+
| 80 | pr74qo-dss-0000 | a2828xx | RW          | USER ACCEPTED       |
| 81 | pr74qo-dss-0000 | di57gix | RW          | USER ACCEPTED       |
+----+-----------------+---------+-------------+---------------------+
```

In the following examples, we will examine invitations for the example user a2828xx.

```plaintext
(dsscli) dss invitation list --username a2828xx
+----+-----------------+---------+-------------+---------------------+
<table>
<thead>
<tr>
<th>ID</th>
<th>Container</th>
<th>User</th>
<th>Access Mode</th>
<th>Status</th>
</tr>
</thead>
</table>
+----+-----------------+---------+-------------+---------------------+
| 80 | pr74qo-dss-0000 | a2828xx | RW          | USER ACCEPTED       |
| 82 | pr74qo-dss-0001 | a2828xx | RW          | USER ACCEPTED       |
+----+-----------------+---------+-------------+---------------------+
```

In order to view the details of a specific invitation, you can use the `dss invitation show` or `dss invitation show id` command. The first one takes the container and username, the later one the invitation id of the invitation to show as mandatory argument:

```plaintext
(dsscli) dss invitation show pr74qo pr74qo-dss-0001 di57gix
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| ID           | 83                               |
| Container    | pr74qo-dss-0001                  |
| User         | di57gix                          |
| Firstname    |                                  |
| Lastname     |                                  |
| Quota GB     | 600                              |
| Quota Files  | 200000                           |
| Access Mode  | RW                               |
| Status       | USER ACCEPT PENDING              |
| Active Task  | None                             |
| Expires At   | 2018-01-04                       |
| Inviter      | a2828yy                          |
```
### 7.3. Modify Access Rights

In the following we show how certain properties of invitations can be changed. Currently you can change the following properties of an existing invitation:

- For invitations that have not yet been accepted:
  - Prolong expiration date
  - Resend the invitation mail
  - The maximum amount of GB, the user is allowed to consume within the container.
  - The maximum number of files, the user is allowed to create within the container.
- For invitations that have already been accepted:
  - The access mode to grant. This can either be **READ_WRITE** or **READ_ONLY**. However, the later can only be used with containers that implement an enforcement policy of **STRICT**.
  - The maximum amount of GB, the user is allowed to consume within the container.
  - The maximum number of files, the user is allowed to create within the container.

[Click here to see how it works in the GUI](#)

#### 7.3.1. Using the GUI

In order to change the access rights of a container user, please navigate to the Container Detail site.

On this detail page, you'll find the **Container User** section.

For each invitation, you'll find four buttons on the left. Depending on the state of the invitation, these buttons may be enabled or disabled.
The button prolongs a still pending invitation
The button resends the invitation mail for a still pending invitation
The button directs you to the invitation modification form
The button deletes the invitation

When you click on the prolong or resend button, the action will be immediately carried out and a System Message will appear that shows you the result of the action:

The invitation update button will take you to the Invitation Update form. Fill out the form and click on the Modify Button.

If the invitation was successfully changed, the UI will take you back to the DSS Container Details site and show you a System Message, that indicates that the invitation has been updated.

Depending on the type of the update, the invitation may go to a CHANGE PENDING state until the update has been carried out on the connected systems.

Click here to see how it works in the CLI

7.3.2. Using the CLI

In order to modify an existing invitation, use the dss invitation update command.

This command takes the following mandatory arguments:

- The ID of the invitation to update

Additionally, you have to specify at least one of the following arguments:

- The new access mode.
- The new maximum usable size.
- The new maximum number of files.
• The prolong flag
• The resend flag

In the following example, we are going to prolong the invitation 83 to be valid for another 30 days:

(dsscli) dss invitation update 83 --prolong
+-------------------------------+
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>83</td>
</tr>
<tr>
<td>Container</td>
<td>pr74qo-dss-0001</td>
</tr>
<tr>
<td>User</td>
<td>di57gix</td>
</tr>
<tr>
<td>Quota GB</td>
<td>None</td>
</tr>
<tr>
<td>Quota Files</td>
<td>None</td>
</tr>
<tr>
<td>Access Mode</td>
<td>RO</td>
</tr>
<tr>
<td>Status</td>
<td>USER ACCEPT PENDING</td>
</tr>
<tr>
<td>Active Task</td>
<td>None</td>
</tr>
<tr>
<td>Expires At</td>
<td>2018-01-04</td>
</tr>
<tr>
<td>Inviter</td>
<td>a2828yy</td>
</tr>
</tbody>
</table>
+-------------------------------+

In the following example, we are going to resend the invitation mail for invitation 83:

(dsscli) dss invitation update 83 --resend
+-------------------------------+
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>83</td>
</tr>
<tr>
<td>Container</td>
<td>pr74qo-dss-0001</td>
</tr>
<tr>
<td>User</td>
<td>di57gix</td>
</tr>
<tr>
<td>Quota GB</td>
<td>None</td>
</tr>
<tr>
<td>Quota Files</td>
<td>None</td>
</tr>
<tr>
<td>Access Mode</td>
<td>RO</td>
</tr>
<tr>
<td>Status</td>
<td>USER ACCEPT PENDING</td>
</tr>
<tr>
<td>Active Task</td>
<td>None</td>
</tr>
<tr>
<td>Expires At</td>
<td>2018-01-04</td>
</tr>
<tr>
<td>Inviter</td>
<td>a2828yy</td>
</tr>
</tbody>
</table>
+-------------------------------+

In the following example, we are going to update the invitation 83:

• New maximum size: 1.000 GB
• New maximum files: 1.000.000
• New access mode: READ_WRITE

(dsscli) dss invitation update 83 --accessmode READ_WRITE --maxgb 1000 --maxfiles 10000000
+-------------------------------+
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>83</td>
</tr>
<tr>
<td>Container</td>
<td>pr74qo-dss-0001</td>
</tr>
<tr>
<td>User</td>
<td>di57gix</td>
</tr>
<tr>
<td>Quota GB</td>
<td>1000</td>
</tr>
<tr>
<td>Quota Files</td>
<td>10000000</td>
</tr>
<tr>
<td>Access Mode</td>
<td>RW</td>
</tr>
</tbody>
</table>
+-------------------------------+
### 7.4. Revoking Access Rights

In the following, we show how existing invitations can be revoked/deleted. You can revoke accepted as well as pending invitations. If you revoke a pending invitation, the invitation link, sent to the person you invited will be invalidated, so accepting the invitation is no longer possible. If you revoke an already accepted invitation, the user will be removed from the container access group and file ownership of each file, technically owned by the user, will be transferred to a data curator. See Understanding Container Access Revocation and User Deletion Actions for more information on the topic.

**Click here to see how it works in the GUI**

#### 7.4.1. Using the GUI

In order to delete the access rights of a container user, please navigate to the Container Detail site.

On this detail page, you'll find the **Container User** section.

For each invitation, you'll find four buttons on the left. Depending on the state of the invitation, these buttons may be enabled or disabled.

- **The** button prolongs a still pending invitation
- **The** button resends the invitation mail for a still pending invitation
- **The** button directs you to the invitation modification form
- **The** button deletes the invitation

In order to delete an invitation, click on the **Delete** Button. After that, a confirmation dialog will appear.

If you confirm the deletion by clicking on the **Delete** Button of the confirmation dialog, the particular invitation will automatically turn into **DELETE PENDING** state and once, deletion is complete will disappear from the **Container User** list.
7.4.2. Using the CLI

In order to revoke access to a container for a particular user, use the `dss invitation delete` command.

Whenever a user, invited to a container will be deleted in the Identity Management System, we will automatically remove the user from the container access group and transfer file ownership of the files, technically owned by that user, to a data curator. (See Understanding Container Access Revocation and User Deletion Actions). As we add other possible actions in the future, we will make this configurable per container.

In the following we are going to delete the invitation for user `di57gix` on container `pr74qo-dss-0001`:

```
(dsscli) dss invitation delete pr74qo-dss-0001 di57gix
Successfully initiated deletion of invitation 83!
```

After this command was issued, the invitation will transition into the `DELETE PENDING` state and will disappear after the file ownership transfer job discussed above, was successfully launched.

8. Managing DSS Container Access Rights for External Users via Globus

If you need to share data with people who do not have a user account at LRZ, LMU or TUM, you can do so, as discussed above, using the Globus Sharing feature of the Globus Data Management Portal.

Please note that you don't have to (and should not) use Globus Sharing if the person you want to share data with has an active LRZ, LMU or TUM account. Globus Sharing is only a fall back solution for sharing data with people who don't have and can't get a valid user account in the mentioned institutions. If possible, always prefer Internal Access Rights.

Please note that in order to manage Globus Sharing ACLs, you first must enable the particular data container for Globus Sharing. See the DSS Data Container section above for how to do this.

Please note that for Globus Sharing, we do not implement an asynchronus task model. Also we do not store any data for Globus Sharing locally. Instead we just "remote control" Globus, using their REST API. Therefore Globus Sharing commands may take longer to execute than other commands.

8.1. Granting Globus Sharing Access Rights

In order to allow a user to access a particular data container, you must invite him or her via Globus Sharing to access the container data. In Globus terms this is called "creating a Globus Sharing ACL".
In order to invite an external user to access a particular data container via Globus Sharing, use the `dss globus invitation create` command.

This command takes the following mandatory arguments:

- The name of the data container
- The E-Mail address of the person to invite

Additionally, you can specify the following optional arguments:

- The access mode to grant. This can either be `READ_WRITE` (the default) or `READ_ONLY`.

In the following example, we are going to invite user alice@external.com to container pr74cu-dss-0001, using:

```
(dsscli) dss globus invitation create --containername pr74cu-dss-0001 --email 'alice@external.com' --accessmode READ_WRITE
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>69b9848a-3f4a-11e8-ba24-0ac6873fc732</td>
</tr>
<tr>
<td>Container</td>
<td>pr74cu-dss-0001</td>
</tr>
<tr>
<td>Subpath</td>
<td>/</td>
</tr>
<tr>
<td>Username</td>
<td><a href="mailto:alice@external.com">alice@external.com</a></td>
</tr>
<tr>
<td>User ID</td>
<td>544f1075-41eb-4ea7-bd97-f418007fa0e6</td>
</tr>
<tr>
<td>Name</td>
<td>None</td>
</tr>
<tr>
<td>Email</td>
<td>None</td>
</tr>
<tr>
<td>User Org</td>
<td>None</td>
</tr>
<tr>
<td>Access Mode</td>
<td>RW</td>
</tr>
<tr>
<td>Status</td>
<td>unused</td>
</tr>
<tr>
<td>Admin Role</td>
<td>None</td>
</tr>
</tbody>
</table>

8.2. Showing Globus Sharing Access Rights

8.2.1. Using the GUI

Click here to see how it works in the GUI

8.2.2. Using the CLI

Click here to see how it works in the CLI

In order to get an overview of all Globus Sharing ACLs to containers of your project via DSSWeb, you can use the `dss globus invitation list` command. Additionally you can limit the output to the invitations of a particular container and/or username.

In the following examples, we will examine invitations for the example project `pr74qo`.

```
(dsscli) dss globus invitation list pr74qo
```

<table>
<thead>
<tr>
<th>ID</th>
<th>Container</th>
<th>User</th>
<th>Access Mode</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>69b9848a-3f4a-11e8-ba24-0ac6873fc732</td>
<td>pr74qo-dss-0001</td>
<td><a href="mailto:alice@external.com">alice@external.com</a></td>
<td>RW</td>
<td>unused</td>
</tr>
</tbody>
</table>

In the following examples, we will examine invitations for the example container `pr74qo-dss-0001`.
In order to view the details of a specific Globus Sharing ACL, you can use the `dss globus invitation show`. The command takes the container name and Globus ACL ID as mandatory arguments:

```bash
(dsscli) dss globus invitation show pr74qo-dss-0001 69b9848a-3f4a-11e8-ba24-0ac6873fc732
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>69b9848a-3f4a-11e8-ba24-0ac6873fc732</td>
</tr>
<tr>
<td>Container</td>
<td>pr74qo-dss-0001</td>
</tr>
<tr>
<td>Subpath</td>
<td>/</td>
</tr>
<tr>
<td>Username</td>
<td><a href="mailto:alice@external.com">alice@external.com</a></td>
</tr>
<tr>
<td>User ID</td>
<td>544f1075-41eb-4ea7-bd97-f418007fa0e6</td>
</tr>
<tr>
<td>Name</td>
<td>None</td>
</tr>
<tr>
<td>Email</td>
<td>None</td>
</tr>
<tr>
<td>User Org</td>
<td>None</td>
</tr>
<tr>
<td>Access Mode</td>
<td>RW</td>
</tr>
<tr>
<td>Status</td>
<td>unused</td>
</tr>
<tr>
<td>Admin Role</td>
<td>None</td>
</tr>
</tbody>
</table>

8.3. Revoking Globus Sharing Access Rights

8.3.1. Using the GUI

Click here to see how it works in the GUI

8.3.2. Using the CLI

In order to revoke access to a container for a particular user, use the `dss globus invitation delete` command. The command takes the container name and Globus ACL ID as mandatory arguments:

```bash
(dsscli) dss globus invitation delete pr74qo-dss-0001 69b9848a-3f4a-11e8-ba24-0ac6873fc732
```

Successfully initiated deletion of globus invitation 69b9848a-3f4a-11e8-ba24-0ac6873fc732!

9. Managing DSS Container Auto Group Links
If you already manage a TUMOnline Group and want that everyone in this group automatically gets an invitation for a particular data container, you can do so using the Container Autor Group Links already discussed above. In the following we will show you how to manage these Auto Group Links.

9.1. Creating Auto Group Links

9.1.1. Using the GUI

9.1.2. Using the CLI

In order to create a new auto group link for a particular container, use the `dss autogrplnk create` command.

This command takes the following mandatory arguments:

- The name of the data container
- The group name to link
- The group origin (The service for which this group is defined)

Additionally, you can specify the following optional arguments:

- The access mode to grant. This can either be `READ_WRITE` (the default) or `READ_ONLY`. However, the later can only be used with containers that implement an `enforcement policy` of `STRICT`.

In the following example, we are going to create a `READ_WRITE` auto group link for container `pr74qo-dss-0003` to group `dssainvtest1`, which is defined in the DSS service.

```
(dsscli) dss autogrplnk create --containername pr74qo-dss-0003 --grouporigin DSS --groupname dssainvtest1 --accessmode READ_WRITE
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>8</td>
</tr>
<tr>
<td>Container</td>
<td>pr74qo-dss-0003</td>
</tr>
<tr>
<td>Group</td>
<td>dssainvtest1</td>
</tr>
<tr>
<td>Group Origin</td>
<td>DSS</td>
</tr>
<tr>
<td>Access Mode</td>
<td>RW</td>
</tr>
<tr>
<td>Created</td>
<td>2018-04-10T10:50:26.752592+02:00</td>
</tr>
<tr>
<td>Last Updated</td>
<td>2018-04-10T10:50:26.752623+02:00</td>
</tr>
</tbody>
</table>

9.2. Showing Auto Group Links

9.2.1. Using the GUI

9.2.2. Using the CLI

In order to get an overview about all existing auto group links via DSSWeb you have access to, you can use the `dss autogrplnk list` command. Additionally you can limit the output to the invitations of a particular project or container.

In the following example, we will examine auto group links for the example project `pr74qo`.

```
(dsscli) dss autogrplnk list --projectname pr74qo
```

<table>
<thead>
<tr>
<th>ID</th>
<th>Container</th>
<th>Group</th>
<th>Group Origin</th>
<th>Access Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In the following example, we will examine auto group links for the example container pr74qo-dss-0003.

```
(dsscli) dss autogrplnk list --containername pr74qo-dss-0003
```

<table>
<thead>
<tr>
<th>ID</th>
<th>Container</th>
<th>Group</th>
<th>Group Origin</th>
<th>Access Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>pr74qo-dss-0003</td>
<td>dssainvtest1</td>
<td>DSS</td>
<td>RW</td>
</tr>
<tr>
<td>9</td>
<td>pr74qo-dss-0003</td>
<td>dssainvtest2</td>
<td>DSS</td>
<td>RW</td>
</tr>
</tbody>
</table>

In order to view the details of a specific auto group link, you can use the `dss autogrplnk show` command. The command takes the auto group link id of the auto group link to show as mandatory argument:

```
(dsscli) dss autogrplnk show 8
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>8</td>
</tr>
<tr>
<td>Container</td>
<td>pr74qo-dss-0003</td>
</tr>
<tr>
<td>Group</td>
<td>dssainvtest1</td>
</tr>
<tr>
<td>Group Origin</td>
<td>DSS</td>
</tr>
<tr>
<td>Access Mode</td>
<td>RW</td>
</tr>
<tr>
<td>Created</td>
<td>2018-04-10T10:50:26.752592+02:00</td>
</tr>
<tr>
<td>Last Updated</td>
<td>2018-04-10T10:50:26.752623+02:00</td>
</tr>
</tbody>
</table>

Please note that invitations from auto group links are created asynchronously. So it may take a while (usually up to one hour) until invitations get created after a new auto group link has been defined.

9.3. Updating Auto Group Links

9.3.1. Using the GUI

9.3.2. Using the CLI

In order to modify an existing auto group link, use the `dss autgrplnk update` command.

This command takes the following mandatory arguments:

- The ID of the auto group link to update
- The access mode to grant. This can either be `READ_WRITE` (the default) or `READ_ONLY`. However, the later can only be used with containers that implement an enforcement policy of `STRICT`.

In the following example, we will change the access mode of auto group link 8 from `READ_ONLY` to `READ_WRITE`.

```
(dsscli) dss autogrplnk update 8 --accessmode READ_WRITE
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>8</td>
</tr>
</tbody>
</table>
9.4. Deleting Auto Group Links

9.4.1. Using the GUI

9.4.2. Using the CLI

In order to delete an existing auto group link, use the dss autogrplnk delete command.

In the following example, we will delete auto group link 8.

(dsscli) dss autogrplnk delete 8
Successfully deleted auto invitation group link 8!

10. Managing DSS Container Exports

Dynamically managing container exports is unfortunately not yet supported because of a technical limitation in the underlying NFS server used. This problem will be addressed and fixed in a future release of DSS. If you need your container exported via NFS to LRZ Cloud Machines or Virtual Machines for example, please open a ticket at the LRZ Service Desk for it. We will setup the exports as soon as possible for you.

10.1. Limitations

Please note that there are several limitations for exporting DSS containers via NFS.

You can only export DSS containers to IPs that are located in the LRZ data center. Exporting to IPs external to LRZ is not supported.

All DSS container exports are configured with root_squash enabled. Therefore it is not possible to have "root access" on the data via an NFS client.

Also technically not forbidden, you should only export DSS containers to IPs that are statically assigned to and trusted by you. NFS exports follow a "host based trust" semantic, which means the DSS NFS server will trust any IP/system to which a DSS container is exported. There is no additional user authentication between NFS server and client enforced. This is especially important if you want to export DSS containers to cloud machines, as these - by default - use a dynamically allocated IP, which may be reused by other machines as soon as you shut down your VM.

10.2. Creating exports

Dynamically managing container exports is unfortunately not yet supported because of a technical limitation in the underlying NFS server used. This problem will be addressed and fixed in a future release of DSS. If you need your container exported via NFS to LRZ Cloud Machines or Virtual Machines for example, please open a ticket at the LRZ Service Desk for it. We will setup the exports as soon as possible for you.
10.3. Showing exports

Click here to see how it works in the GUI

10.3.1. Using the GUI

In order to display the NFS exports of a container user, please navigate to the Container Detail site.

On this detail page, you'll find the **NFS Exports** section:

![Screencap of DSSWeb showing Container Detail and NFS Exports](image)

Click here to see how it works in the CLI

10.3.2. Using the CLI

In order to get an overview about all NFS exports of containers of your project via DSSWeb, you can use the `dss nfsexport list` command. Additionally you can limit the output to the exports of a particular container and/or pool name.

In the following examples, we will examine NFS exports for the example project `pr74qo`.

```plaintext
(dsscli) dss nfsexport list pr74qo

<table>
<thead>
<tr>
<th>ID</th>
<th>Container</th>
<th>IP</th>
<th>Access Mode</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>pr74qo-dss-0002</td>
<td>10.156.29.72</td>
<td>RW</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>23</td>
<td>pr74qo-dss-0002</td>
<td>10.156.29.73</td>
<td>RW</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>24</td>
<td>pr74qo-dss-0001</td>
<td>10.156.29.74</td>
<td>RW</td>
<td>ACTIVE</td>
</tr>
</tbody>
</table>
```
In order to view the details of a specific NFS export, you can use the command. The command takes the NFS export id of the export to show as mandatory argument:

```
(dsscli) dss nfsexport show 22
+-------------+------------------------------------------------------------------+
| Field       | Value                                                            |
+-------------+------------------------------------------------------------------+
| ID          | 22                                                               |
| Container   | pr74qo-dss-0002                                                  |
| IP          | 10.156.29.72                                                     |
| Access Mode | RW                                                               |
| Status      | ACTIVE                                                           |
| Mount Path  | datdsscest102.dss.lrz.de:/dss/dsstestfs01/pr74qo/pr74qo-dss-0002 |
| Expires     | None                                                             |
| Active Task | None                                                             |
| Created     | 2017-12-06T08:07:40.997646+01:00                                 |
| Last Update | 2017-12-06T08:07:51.507205+01:00                                 |
+-------------+------------------------------------------------------------------+
```

10.4. Deleting exports

Dynamically managing container exports is unfortunately not yet supported because of a technical limitation in the underlying NFS server used. This problem will be addressed and fixed in a future release of DSS. If you need your container unexported via NFS
11. Hints and possible pitfalls

11.1. Known Limitations

11.2. Do's and Dont's