Configure B2SAFE

About

Documentation how to configure the B2SAFE service for Site Administrators and Community Data Managers.

Modified: 12 February 2018

Synopsis

B2SAFE, the EUDAT Safe Replication service, is the cornerstone of EUDAT's service for safe replication. In addition to replication, the service also allows communities to implement data management policies on their research data across multiple administrative domains in a trustworthy manner.

This document describes how to implement and configure B2SAFE such that a community data centre can join EUDAT.

The document applies to B2SAFE version 4.0.1.

Acronyms

PID: Persistent identifier associated to a file or iRODS collection, usually an EUDAT Handle.

ROR: Repository of Records, (persistent) identifier to the original data object. Can be of any identifier type. If the community does not assign an own identifier the B2SAFE PID will be used.

FIO: First ingested object, the persistent identifier associated to the very first object in in the EUDAT domain. If the the chain has only two elements, the master copy and the first replica, then the PARENT = FIO.

PARENT: Parent PID, the persistent identifier associated to the source object in a replication chain. If the chain has only two elements, the master copy and the first replica, then the PARENT = FIO.

Digital entity: Files and folders

Digital object: Files and folders that carry a persistent identifier and possibly some metadata.

What is B2SAFE

The EUDAT B2SAFE Service offers the functionality to replicate datasets across different data centres in a safe and efficient way while maintaining all information required to easily find and query information about the replica locations. To this end information on locations along with other important information is stored in PID records. The B2SAFE Service is implemented as an iRODS module providing a set of iRODS rules or policies to interface with either the Handle HTTP JSON REST API or the EPIC Handle API to create and manage PIDs and uses the iRODS middleware to replicate datasets from a source data (or community) centre to a destination data centre. B2SAFE offers the possibility to replicate single files (also called data objects) and set of data objects organised in iRODS collections. The organisation of data is left to the communities. While B2SAFE is internally using the handle HTTP JSON REST API or the EPIC Handle API, communities have the choice to use any PID system they prefer to assign PIDs to their digital objects at the ROR. In any case, replicas in the EUDAT domain will receive a handle upon

EUDAT receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 654065.

replication by the B2SAFE service. The PIDs are returned by the B2SAFE service and can be employed to reference datasets. For more information of B2SAFE please read our B2SAFE user documentation page.

Be aware that the EPIC API is being fazed out in favor of the handle HTTP JSON REST API.

Communities can elect to ask an EUDAT site to run B2SAFE for them; please see our Using B2SAFE guide for more information. This guide is for communities that have elected to run their own B2SAFE service, and it explains how to achieve this.

In the following we will provide a comprehensive guide to install the basic features of B2SAFE. For more information and further configurations please see the B2SAFE wiki.

Configure B2SAFE

This section explains the required steps for a **Community Centre Administrator** to configure the B2SAFE Service at their site.

Prerequisites

Before installing and configuring the EUDAT B2SAFE module, make sure you meet the following conditions:

- You are running an iRODS server, either version 4.0.X, **4.1.X (preferred)** or 4.2.X (User Documentation - iRODS Deployment)
- You have a Handle prefix (User Documentation - PIDs in EUDAT) with installed private public keys and certificates (see B2HANDLE documentation); you can create and update persistent identifiers with this prefix (please test before installing B2SAFE)
- You have access to the EUDAT git repository
- You have access to the EUDAT B2HANDLE git repository to create and install the B2HANDLE library
- You have enabled access to ports 1247, 1248 and 20000-20199 that iRODS uses to communicate.

Definitions

- `<irods>` denotes the absolute path to your iRODS directory. e.g `/var/lib/irods/iRODS`
- `<irods user>` unix service account which runs iRODS; iRODS creates such an account and calls it by default "irods"
- `<b2safe-download>` denotes the absolute path to your B2SAFE module downloaded from git. e.g `/home/<user>/B2SAFE-core` or `/var/lib/irods/B2SAFE-core`
- `<B2SAFE-core>` Configuration folder, usually `/opt/eudat/b2safe`

iRODS Versions Supported

B2SAFE module release 3.X.X needs to be installed with iRODS 4.0.X or 4.1.X. In the following we describe the installation of B2SAFE through packages first.

The installation through packages is the preferred way of installing B2SAFE.

Steps

EUDAT receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 654065.

0. Get and install the B2HANDLE library

B2SAFE uses Handles as Persistent Identifiers and it manages them using the B2HANDLE python library.

To install the B2HANDLE python library download the latest version from github and follow the installation guide (installation via a python egg file). You may also create an rpm as follows and install it:

```
python setup.py bdist_rpm
rpm -ivh <path>/dist/b2handle-<version>.noarch.rpm
```

Please also make sure you have installed the following python packages:

```
queuelib, lxml, defusedxml, httplib2 and simplejson
```

If you have not done so beforehand, please generate private/public keys for your Handle prefix following the instructions (you will need to download the Handle software) and install or let the system administrator of your local Handle server install them for you.

Before you proceed with the installation of B2SAFE, please make sure that you can create and update Persistent Identifiers with your prefix like demonstrated in the tutorial.

1. Get the B2SAFE module

From https://github.com/EUDAT-B2SAFE/B2SAFE-core/releases you may download a B2SAFE release that is suitable for you. When you download the B2SAFE zip file and unzip it (as a **NOT root** user), it creates a directory named "B2SAFE-core-4.X.X". You may rename this directory, but in this documentation we refer to it as `<b2safe-download>` . The directory name must be different to any iRODS directories.

2. Installation through Binary Packages

2.1. RPM

Go to the directory where the packaging files are:

```
cd <b2safe-download>/packaging
```

Create package:

```
./create_rpm_package.sh
```

Login as root and install package:

```
rpm -ivh /home/<user>/rpmbuild/RPMS/noarch/irods-eudat-b2safe-3.X-X.noarch.rpm
  Preparing...                ########################################### [100%]
     1:irods-eudat-b2safe     ########################################### [100%]
```

2.2. DEB

Go to the directory where the packaging files are:

EUDAT receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 654065.

```
cd <b2safe-download>/packaging
```

Create the package

```
./create_deb_package.sh
```

You should see an output like:

```
dpkg-deb: building package `irods-eudat-b2safe' in `/home/<user>/debbuild/irods-eudat-b2safe_3.X-X.deb’
```

Login as root or use sudo to install package:

```
root# dpkg -i /home/<user>/debbuild/irods-eudat-b2safe_3.X-X.deb
Selecting previously unselected package irods-eudat-b2safe.
(Reading database ... 48938 files and directories currently installed.)
Unpacking irods-eudat-b2safe (from .../irods-eudat-b2safe_3.X-X.deb) ...
Setting up irods-eudat-b2safe (3.X-X) ...
```

2.3. Edit the Configuration File and Run the Installation Script

In the configuration file you define several parameters setting the location of iRODS and B2SAFE, setting some iRODS-specific parameters and setting the Handle authentication.

The package b2safe is installed in `/opt/eudat/b2safe`. To configure B2SAFE adjust the install.conf file as follows:

```
sudo vi /opt/eudat/b2safe/packaging/install.conf
```

Please make sure you set the parameters `MSIFREE_ENABLED` and `MSICURL_ENABLED` to false for a default installation. If you set them to true you will be required to install some extra iRODS patches (see Section 2.3.2).

EUDAT no longer supports the ePIC API; please follow the Handle System REST API (employed by the B2HANDLE python library) instructions in 2.3.1 below.

2.3.1 Settings for the Handle HTTP JSON REST API

- `SERVER_ID`: The fully qualified name of your iRODS server or its IP address
- `HANDLE_SERVER_URL`: The Handle server that hosts your Handle prefix with the port
- `PRIVATE_KEY`: The private key to authenticate with at the Handle server
- `CERTIFICATE_ONLY`: The certificate which contains the DN used to authenticate against
- `PREFIX`: Your prefix
- `HANDLE_OWNER`: The owner of the handles being created. Normally something like: "200:0.NA/<PREFIX>"
- `REVERSELOOKUP_USERNAME`: The username to use when doing a reverse lookup
- `HTTPS_VERIFY`: Describes whether to check the certificate of the Handle HTTP JSON REST API server. Default "True"

The `SERVER_ID` is used to build the URL when labelling data in iRODS with PIDs, which becomes important when you want to retrieve data with gridFTP. Then the `SERVER_ID` has to match the name of the gridFTP server.

EUDAT receives funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 654065.

At the end of this file you will find

```
AUTHZ_ENABLED=<false, true>
MSG_QUEUE_ENABLED=<false, true>
```

These are flags for certain plugins you would have to install.

- AUTHZ: makes use of authorisation rules to execute external scripts. Please follow the instruction in Section 3.8. Enabling Authorization.
- MSG_QUEUE_ENABLED: Use messaging (see 'Experimental features' below)

For more detail on these specific configurations see the B2SAFE wiki for rule set configurations.

Then run:

```
source /etc/irods/service_account.config
```

Login as the user who runs iRODS (usually irods or rods) and do:

```
cd /opt/eudat/b2safe/packaging
./install.sh
```

Before running B2SAFE make sure you have installed the following python packages:

```
queuelib, lxml, defusedxml, httplib2 and simplejson
```


When the B2SAFE module is properly installed you should be able to test the rules in <b2safe-download>/rules:

```
eudat-b2safe:~/B2SAFE-core$ irule -F rules/eudatGetV.r
*version=4.1-0
```

To test the creation of PIDs do the following:

- create a file in iRODS

```
eudat-b2safe:~/B2SAFE-core$ echo "test" > testfile.txt
eudat-b2safe:~/B2SAFE-core$ iput testfile.txt
```

- create rule eudatPidsColl.r

```
eudatPidsColl{
    # Create PIDs for all collections and objects in the collection recursively
    # Parameters: *parent_pid, *source, *ror, *fio, "true"(fixed content) , *newPID
    EUDATCreatePID("None", *coll_path, "None", "None", "true", *newPID);
```

EUDAT receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 654065.

```
}
INPUT *coll_path='/<zone>/home/<user>/<file or collection>'
OUTPUT *newPID,ruleExecOut

eudat-b2safe:~/B2SAFE-core$ irule -F rules/eudatPidsColl.r
*newPID = 846/839b44e8-7678-11e6-b7b4-0200c0a809d6
eudat-b2safe:~/B2SAFE-core$ imeta ls -d testfile.txt
```

You can test the replication with the rule <b2safe-download>/rules/eudatRepl:

```
eudatRepl{
    # Data set replication
    # registered data (with PID registration) (3rd argument - 1st bool("true"))
    # recursive (4th argument 2nd bool("true"))
    EUDATReplication(*source, *destination, bool("true"), bool("true"), *response)
}
INPUT *source='/<your zone>/home/<user>/<collection>', *destination='/<remote
zone>/<user>#<your zone>/<collection>'
OUTPUT ruleExecOut

eudat-b2safe:~/B2SAFE-core$ irule -F rules/eudatRepl.r
eudat-b2safe:~/B2SAFE-core$ ils /<remote zone>/<user>#<your zone>/<collection>
eudat-b2safe:~/B2SAFE-core$ imeta ls -C /<remote zone>/<user>#<your zone>/<collection>
```

In some cases the replication fails. The rule uses a server-to-server replication and expects all known iRODS servers to be configured in `/etc/irods/hosts_config.json`, defining local and remote iRODS instances.

To replicate files between different iRODS zones, you need to set up an iRODS federation, and add the information on the remote iRODS to the attribute "federation" in `/etc/irods/server_config.json`.

```
"federation": [
    {
        "icat_host": "",
        "zone_name": "",
        "zone_key": "",
        "negotiation_key": ""
    }]
```

For more details on how to set up the federation please refer to the iRODS manual.

B2SAFE and other Persistent Identifier Systems

B2SAFE uses by default Handles and the Handle System to label data in B2SAFE and track replicas across the different iRODS instances.

Some communities already have an own persistent identifier system and might want to replace the usage of Handles in B2SAFE with their own System. In such a case the community is responsible for creating scripts that introduce the relation, i.e. the links to replicas and parents, for the data objects. To this end they will need to device new iRODS rules. In order to set this up you need to create an iRODS rule which gets a `sourcePID` and

EUDAT receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 654065.

`replicaPID` as parameters. The logic required to implement this policy depends on the community requirements, the repository and the persistent identifier system. This is a community effort, but EUDAT will provide support if needed.

Additional Tools and Feature Preview

Repository Package

The Repository Package allows communities to ingest data from their own repositories into B2SAFE. The source code and documentation can be found on the EUDAT B2SAFE github page. For now, it is possible to ingest data stored on a file system or in the DSPACE repository. The connectors for FEDORA and ePRINTS are planned to be implemented.

Messaging

Messaging is an experimental feature that provides the results in case of asynchronous (server side triggered) replication process (see the Workflows section). The messages are posted to a queue which can be accessed via an HTTP interface. In the current implementation the queue is provided by dweet.io. The messaging is disabled by default. In order to enabled it, just change from `"false"` to `"true"` the parameter `"*enabled"` in the `"getMessageParameters"` rule in the local.re rule set as follows:

```
getMessageParameters(*msgConfPath, *enabled) {
    *msgConfPath="/var/log/irods/msgManager.conf";
    *enabled=bool("false");
}
```

When enabled the results of the PID registration and data replication workflows will be posted to a dweet url based on the iRODS zone and the username as follows: `http://dweet.io/`*`iRODS_zone_name/username`*

Metadata

Metadata management is an experimental feature. It is disabled by default. When enabled it provides a set of metadata properties for each data object, storing them in a file located as follows: given the object `/path/to/myobject`, the metadata file is stored in `/path/to/.metadata/myobject_metadata.json`. In order to enable metadata, just edit the `"getMetaParameters"` rule in the `local.re` rule set and change the parameter `"*enabled"` from `"false"` to `"true"`.

```
getMetaParameters(*metaConfPath, *enabled) {
    *metaConfPath="/srv/irods/current/modules/B2SAFE/cmd/metadataManager.conf";
    *enabled=bool("false");
}
```

The expected content is as follows:

```
{
  "checksum": "sha2:nmDjK/7k1D5jjMUFoWHjX5qZmke9vpQbR6FaY9sk6eI=",
  "ror": "None",
```

EUDAT receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 654065.

```
  "pid": "842/6cff9eb8-47ef-11e5-a889-fa163e62896a",
  "checksum_timestamp": "01440065311"
}
```

and it is triggered by the rules `EUDATCreatePID` and `EUDATeCHECKSUMupdate`.

Support

Our B2SAFE presentation discusses data replication.

You can access B2SAFE hands-on training material from our github; note in particular the sessions dedicated to configuring B2SAFE and using policies to manage data across iRODS zones.

Support for B2SAFE is available via the EUDAT ticketing system through the webform.

If you have comments on this page, please submit them though the EUDAT ticketing system.

Document Data

Version: 3.9

Authors:

Elena Erastova, elena.erastova@mpcdf.mpg.de

Claudio Cacciari, c.cacciari@cineca.it

Robert Verkerk, robert.verkerk@surfsara.nl

Pascal Dugénie dugenie@cines.fr

Editors:

Kostas Kavoussanakis, k.kavoussanakis@epcc.ed.ac.uk

Themis Zamani, themis@admin.grnet.gr

Christine Staiger, christine.staiger@surfsara.nl

Read more

EUDAT receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 654065.