



Integration of CERNBox in the DBOD service

Abel Cabezas Alonso

08 December 2021

Presentation content

- **1. Brief introduction to the Database On Demand service.**
- **2. Why do we opted for an integration with CERNBox?**
- **3. Integration of CERNBox in the DBOD service**
- **4. Results and Conclusions**

1. Brief introduction to the DBOD Service.

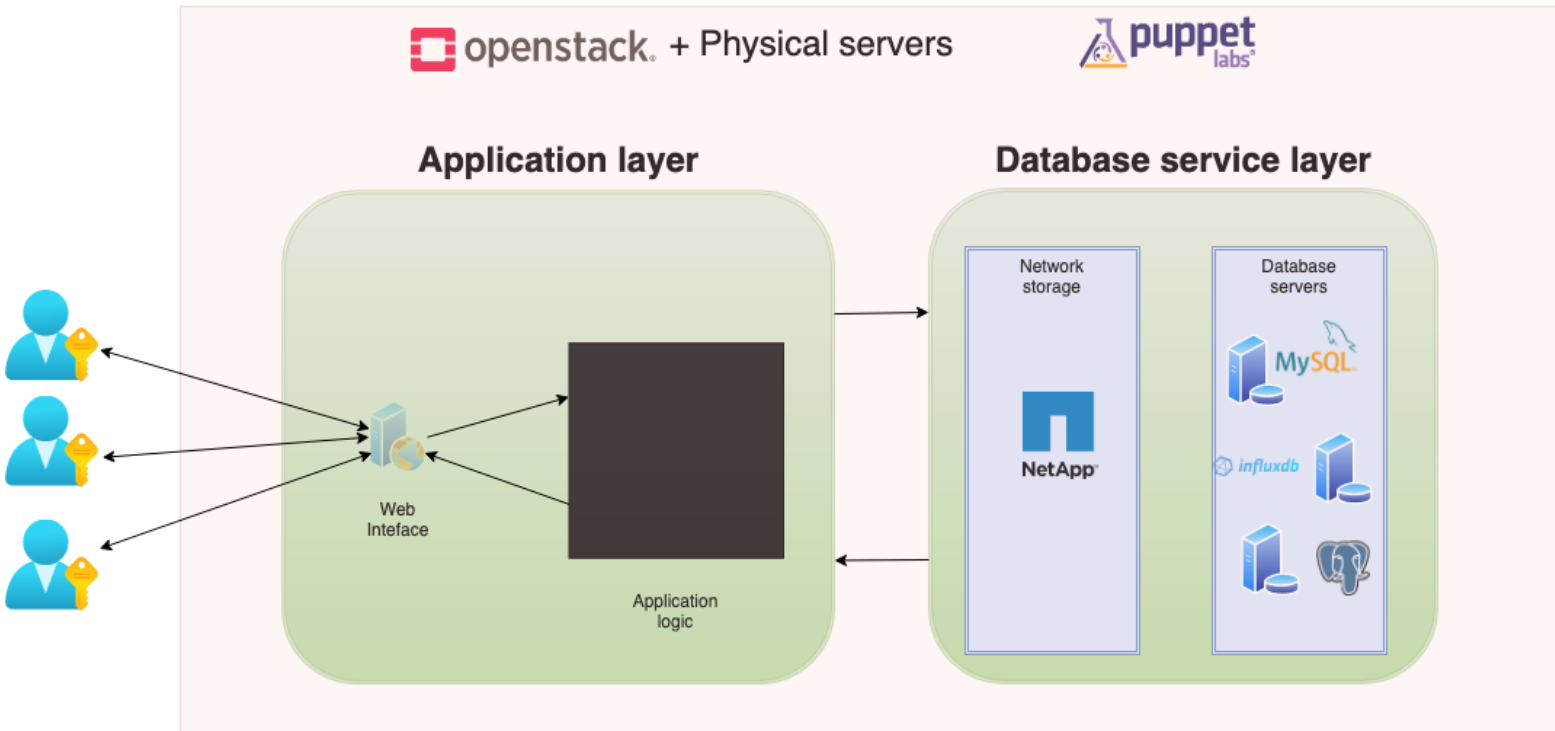
What is the DBOD Service?

- **Born in 2012 was originally conceived as DBaaS (database as a service) with the goal of centralising and standardising procedures for the existing MySQL and PostgreSQL databases within CERN**
- **Provide a free open source alternative to the central Oracle-based database service**
- **The DBOD empowers users to perform certain actions that are traditionally done by DBAs granting them full DBA privileges**
- **Instances can be managed through a simple and intuitive web interface**
- **Any user with a CERN user account can request a DBOD instance**

Database on Demand service architecture

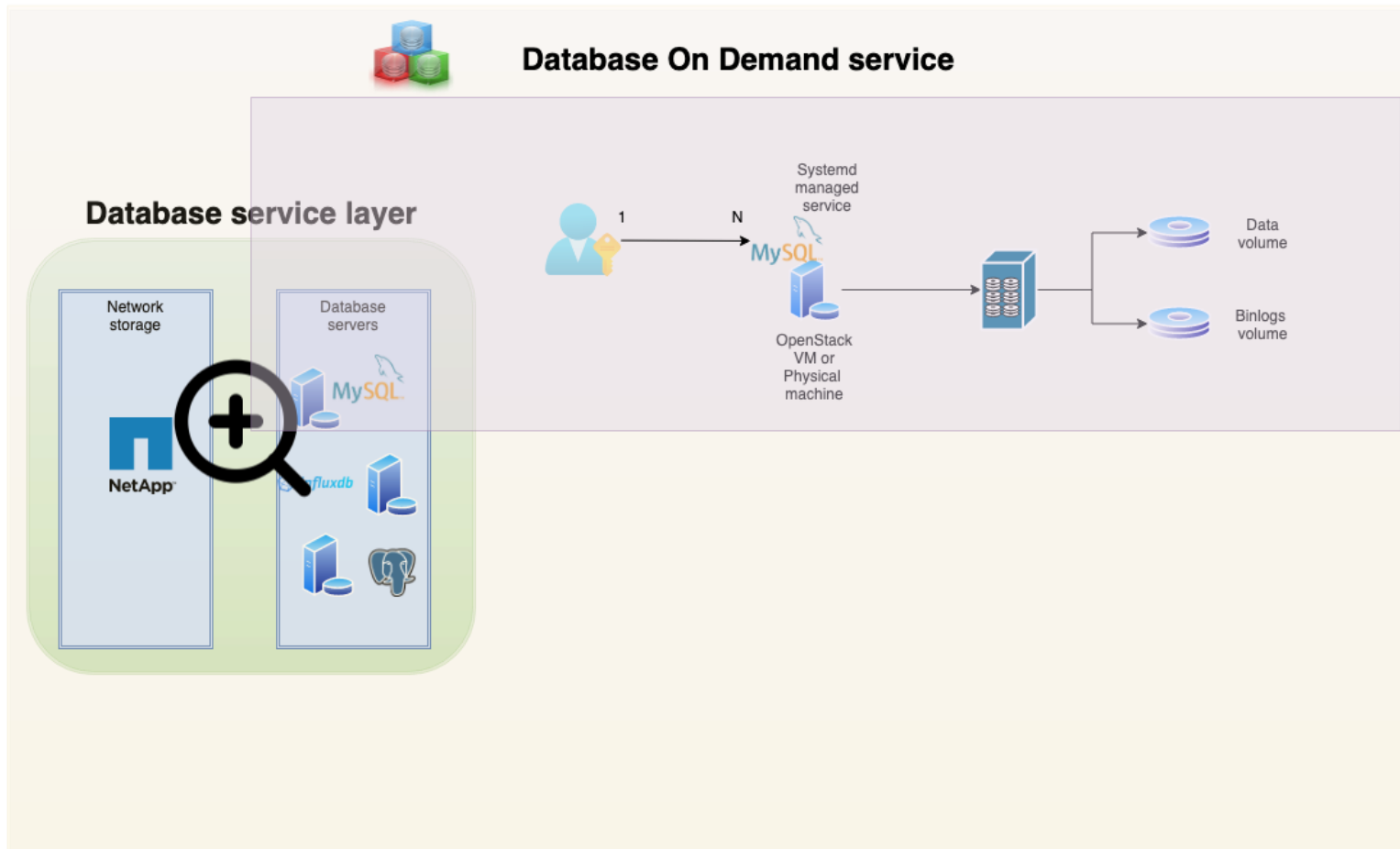


Database On Demand service



- The user interacts directly with the instance as she/he would do in a regular installation
- Web interface for managing start/stop, backups, restoring, monitoring, configuration files, upgrades...
- Puppet managed nodes
- Database servers running on CERN CentOS 7
- Clustered storage network provided by NetApp

Database on Demand service architecture



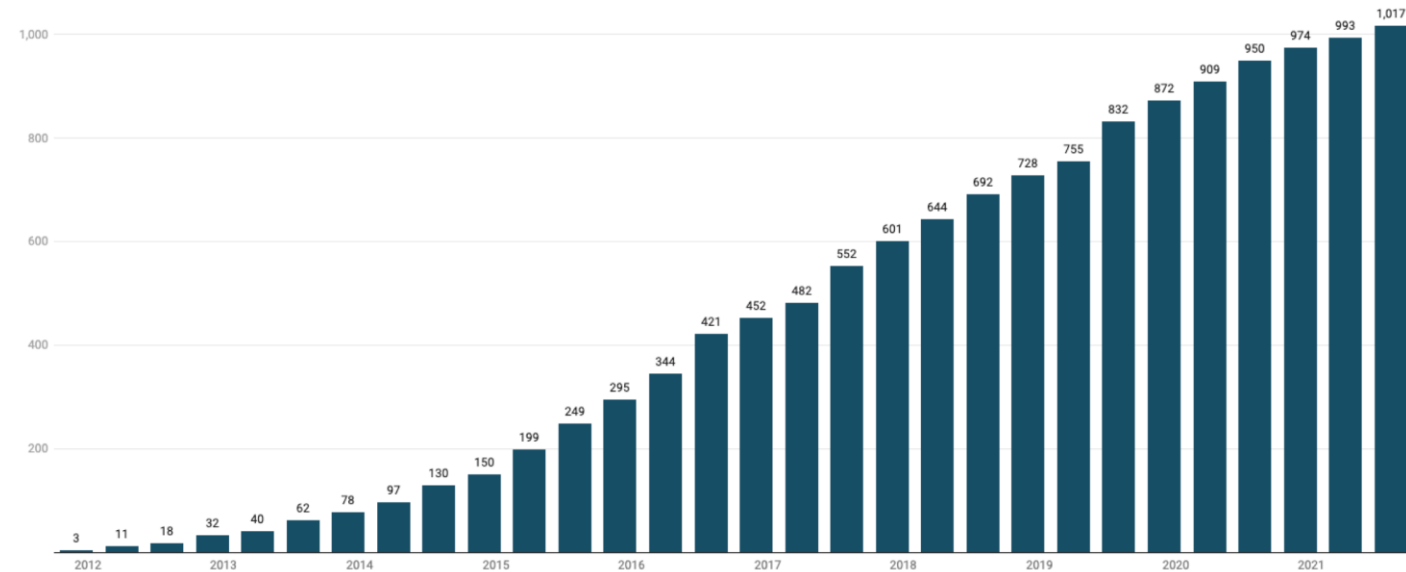
- One user can own from 1 to N instances
- An instance can be owned by only one person but managed by all members of the designated e-group
- Instances run on premise physical or cloud hosts
- A server can host several instances and be run on different Availability Zones
- Each instance's storage is split into two separate volumes

What we provide:

- **Automated backup and recovery services**
- **Monitoring of instance, server and storage metrics**
- **Three different DBMS : MySQL, PostgreSQL and InfluxDB**
- **Automated database upgrades**
- **Cloning**
- **Replication**
- **High availability for MySQL instances**
- **Warranty of service continuity**

The DBOD in numbers

- More than 1000 instances as of December 2021
 - 628 MySQL
 - 348 PostgreSQL
 - 163 InfluxDB
- 158 managed Virtual + Physical hosts
- Around 36 TB of data managed
- Around 800 users subscribed to the service
- 9 years of operational experience



2. Why did we integrated the DBOD service with CERNBox?

Integration with CERNBox, why?

- **Log files traditionally shared through the web interface**
- **Only small files could be downloaded from the web (< 6 MB)**
- **Big log files needed to be requested through a SNOW ticket**
- **Even small files were causing issues in our API**
- **Emerging need to share upgrade checker reports with instance owners and egroup**
- **Recurrent method for sharing files with users**
- **Automation was required**

3. Integration of CERNBox in the DBOD service

DBOD web integration

The screenshot displays the DBOD web interface. At the top left is the DBOD logo (Database on Demand). At the top right, there is a '+ REQUEST NEW INSTANCE' button and the user is signed in as 'Abel Cabezas Alonso from CERN' with a 'Signout' link.

The main content area shows the details for instance 'acabezas_mysql_002'. A dropdown menu is open, showing 'upgrade to version 8.0.23' and 'upgrade checker reports'. A green arrow points to the 'upgrade checker reports' option with the text 'Access upgrade checker reports shared folder'. Below this, a table lists instance details:

| Owner | Egroup | Project | Type | Version | Category | Charge Group | Port | Host | Creation date |
|----------|------------------|---------|-------|---------|----------|--------------------|------|------------|---------------------|
| acabezas | dbondemand-admin | dbod | MYSQL | 8.0.16 | REF | Database on Demand | 5546 | dbod-cr006 | 2020-08-19T00:00:00 |

Below the table, there is an 'Expiry date' field and an 'Extend six months' button.

The 'Logs' tab is selected, showing a list of log messages. A green arrow points to the 'Logs' tab with the text 'Access shared logs folder'. Another green arrow points to the 'Submit latest logs to EOS' button with the text 'Submit latest logs to EOS'. The log messages are as follows:

| Date | Message |
|----------------------|--|
| 3.12.21 14:15:46:168 | [MY-011323] [Server] X Plugin ready for connections. Socket: '/tmp/mysqlx.sock' bind-address: '::' port: 33060 |
| 3.12.21 14:15:46:099 | [MY-010931] [Server] /usr/local/mysql/mysql-8.0.16/bin/mysqld: ready for connections. Version: '8.0.16' socket: '/var/lib/mysql/mysql.sock.acabezas_mysql_002.5546' port: 5546 MySQL Community Server - GPL. |
| 3.12.21 14:15:45:941 | [MY-010068] [Server] CA certificate /etc/dbod/certificates/ca.pem is self signed. |
| 3.12.21 14:15:34:718 | [MY-010116] [Server] /usr/local/mysql/mysql-8.0.16/bin/mysqld (mysqld 8.0.16) starting as process 5743 |
| 3.12.21 14:15:34:717 | [MY-011068] [Server] The syntax 'expire-logs-days' is deprecated and will be removed in a future release. Please use binlog_expire_logs_seconds instead. |
| 3.12.21 14:12:56:282 | 2021-11-29T09:18:25.550908Z mysqld_safe mysqld from pid file /ORA/dbs03/ACABEZAS_MYSQL_002/mysqld/dbod-cr006.cern.ch.pid ended |
| 3.12.21 14:12:56:282 | 2021-11-29T09:18:30.032128Z mysqld_safe Logging to '/ORA/dbs03/ACABEZAS_MYSQL_002/mysqld/dbod-cr006.cern.ch.err'. |
| 3.12.21 14:12:56:282 | 2021-11-29T09:18:30.077443Z mysqld_safe Starting mysqld daemon with databases from /ORA/dbs03/ACABEZAS_MYSQL_002/mysql |
| 3.12.21 14:12:56:282 | 2021-11-29T09:19:11.165222Z mysqld_safe mysqld from pid file /ORA/dbs03/ACABEZAS_MYSQL_002/mysqld/dbod-cr006.cern.ch.pid ended |
| 3.12.21 14:12:56:275 | Version: '5.7.26-log' socket: '/var/lib/mysql/mysql.sock.acabezas_mysql_002.5546' port: 5546 MySQL Community Server (GPL) |

At the bottom left, it says 'DBOD Web | CERN'. At the bottom right, there are links for 'User Guide', 'SNOW Incident', and 'SNOW Request'.

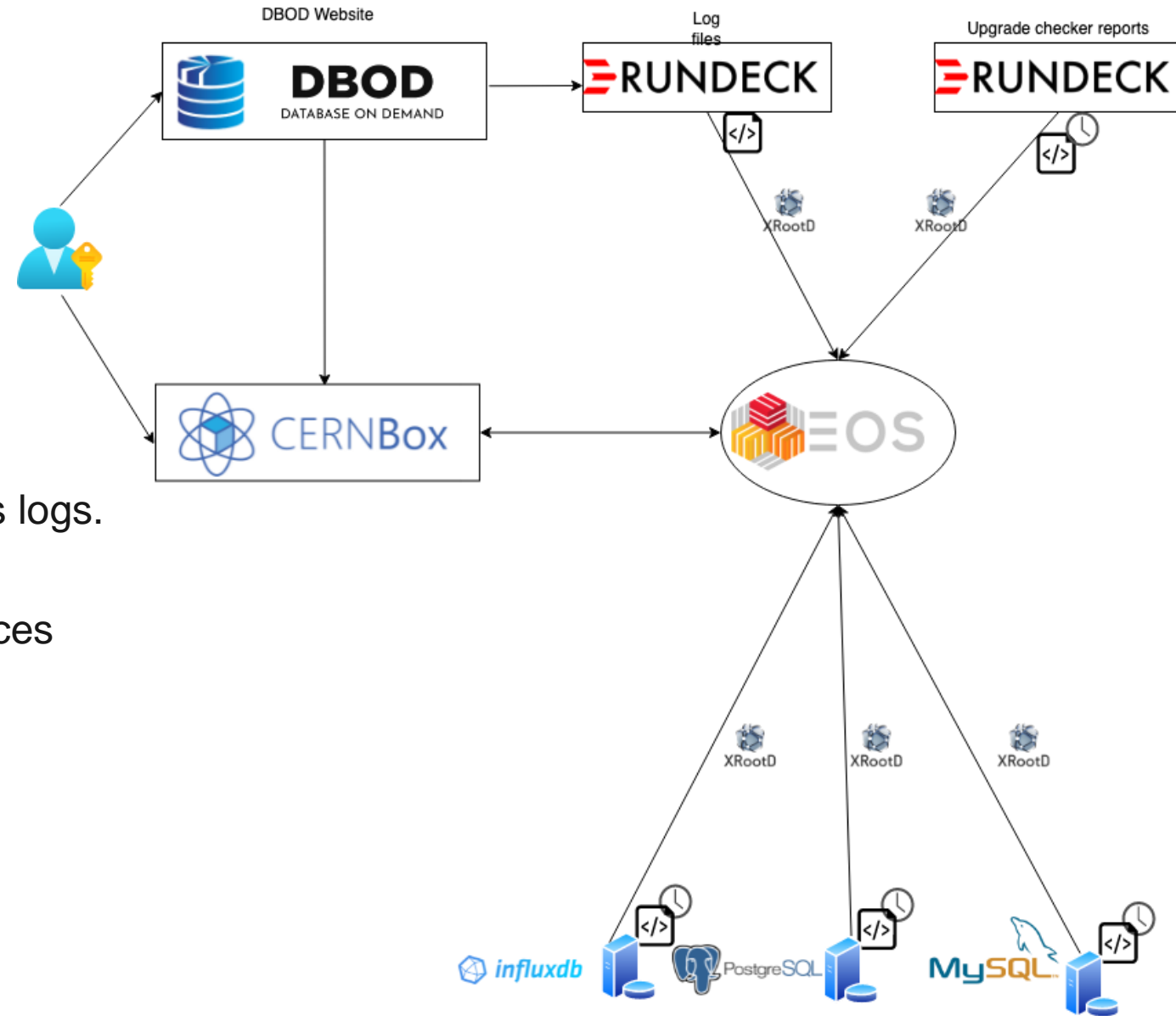
DBOD web integration

The screenshot displays the DBOD (Database On Demand) web interface. At the top left, the DBOD logo and 'DATABASE ON DEMAND' text are visible. The main content area shows a file explorer for the instance 'acabezas_mysql_002' (ID: 411085). The breadcrumb path is '_myshares > acabezas_mysql_002 (id:411085)'. A message states: 'You don't have permission to upload or create files here'. The file explorer shows two folders: 'upgrade_checker_reports' and 'logs'. Green arrows point from the 'upgrade_checker_reports' folder in the left sidebar to the 'upgrade_checker_reports' folder in the main view, and from the 'logs' folder in the left sidebar to the 'logs' folder in the main view. Below the file explorer, there is a 'Jobs' tab and a 'Logs' tab. The 'Logs' tab is active, showing a table of log entries.

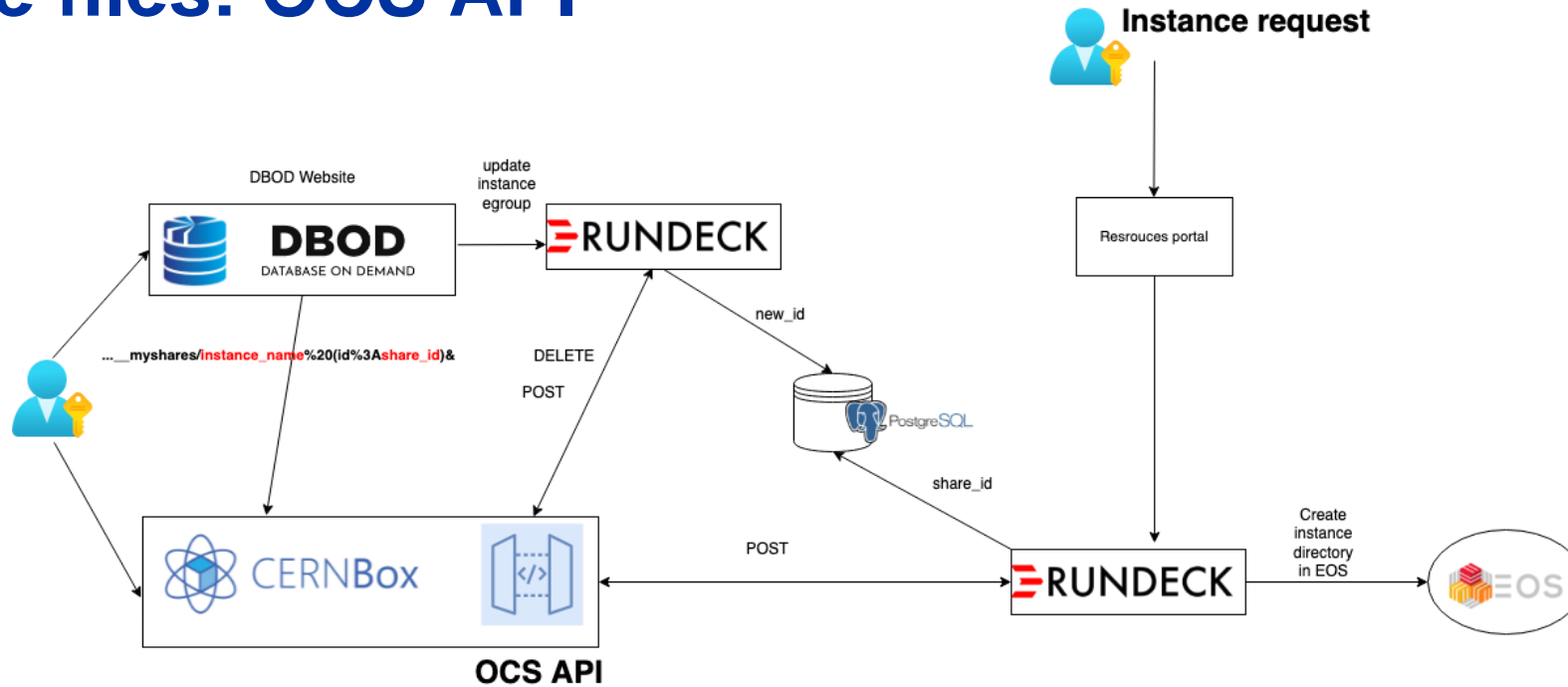
| Date | Message |
|----------------------|--|
| 7.12.21 12:03:46:021 | [MY-010931] [Server] /usr/local/i |
| 7.12.21 12:03:45:888 | [MY-010068] [Server] CA certifica |
| 7.12.21 12:03:45:888 | [MY-013602] [Server] Channel m |
| 7.12.21 12:03:45:629 | [MY-013180] [Server] Function 'r |
| 7.12.21 12:03:45:629 | [MY-010736] [Server] Couldn't load plugin named 'rpl_semi_sync_master' with soname 'semisync_master.so'. |

File uploads

- Log Uploads
 - Hourly
 - Script run as a cron job.
 - From the web:
 - Call rundeck job that uploads the latests logs.
- Upgrade checker reports
 - Upgrade checker is run for all of the instances every 3 days -> result is uploaded
- Use of xrdcp protocol
- Only files that changed are updated



Sharing the files: OCS API



- On instance creation:
 - Create instance directories in EOS
 - Share instance directories with egroup using OCS API
 - Store EOS `share_id` in our internal database (unique `share_id` per instance)
- If user updates instance's egroup:
 - Update existing `share_id` in our database with the `share_id` of the new egroup

Form of the requests - POST

- Create share

```
curl -X POST
```

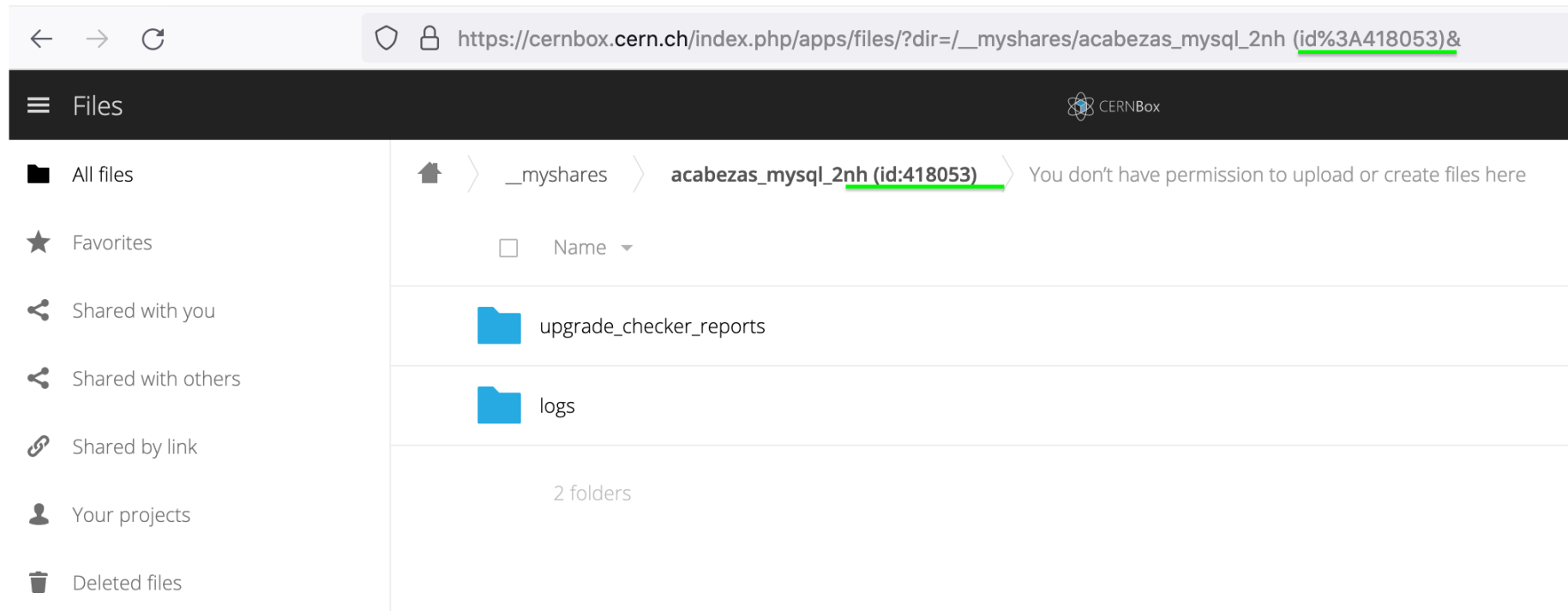
```
'https://cernbox.cern.ch/cernbox/desktop/ocs/v1.php/apps/files_sharing/api/v1/shares?format=json' --data 'shareType=1&shareWith=dbondemand-admin&permissions=1&path=%2F__myprojects%2Fdbod%2Fpublic%2Finstance_name' -u mysql -i
```

- *shareType*:
 - 0 -> user share
 - 1 -> egroup share
- *shareWith*:
 - Name of the user or egroup with which the resource should be shared
- *permission*:
 - 1 -> read
 - 15 -> read/write
- *path*:
 - the path of the resource to share

Form of the requests - POST - Response

- Response from the OCS API:

```
{'ocs': {'meta': {'status': 'ok', 'statusCode': 200, 'message': '', 'totalitems': '', 'itemsperpage': ''}, 'data': {'id': '418053', 'share_type': 1, 'uid_owner': 'mysql', 'displayname_owner': 'mysql', 'permissions': 1, 'stime': 1638550447, 'token': '', 'uid_file_owner': 'mysql', 'displayname_file_owner': 'mysql', 'path': '/_myprojects/dbod/public/acabezas_mysql_2nh', 'item_type': 'folder', 'mimetype': 'httpd/unix-directory', 'item_source': 'projects:30280278', 'file_source': 'projects:30280278', 'file_target': '/_myprojects/dbod/public/acabezas_mysql_2nh', 'file_parent': '', 'share_with': 'dbondemand-admin', 'share_with_displayname': 'dbondemand-admin', 'name': '/_myprojects/dbod/public/acabezas_mysql_2nh', 'url': '', 'state': 0}}}
```



Form of the requests - DELETE & UPDATE

- Delete share

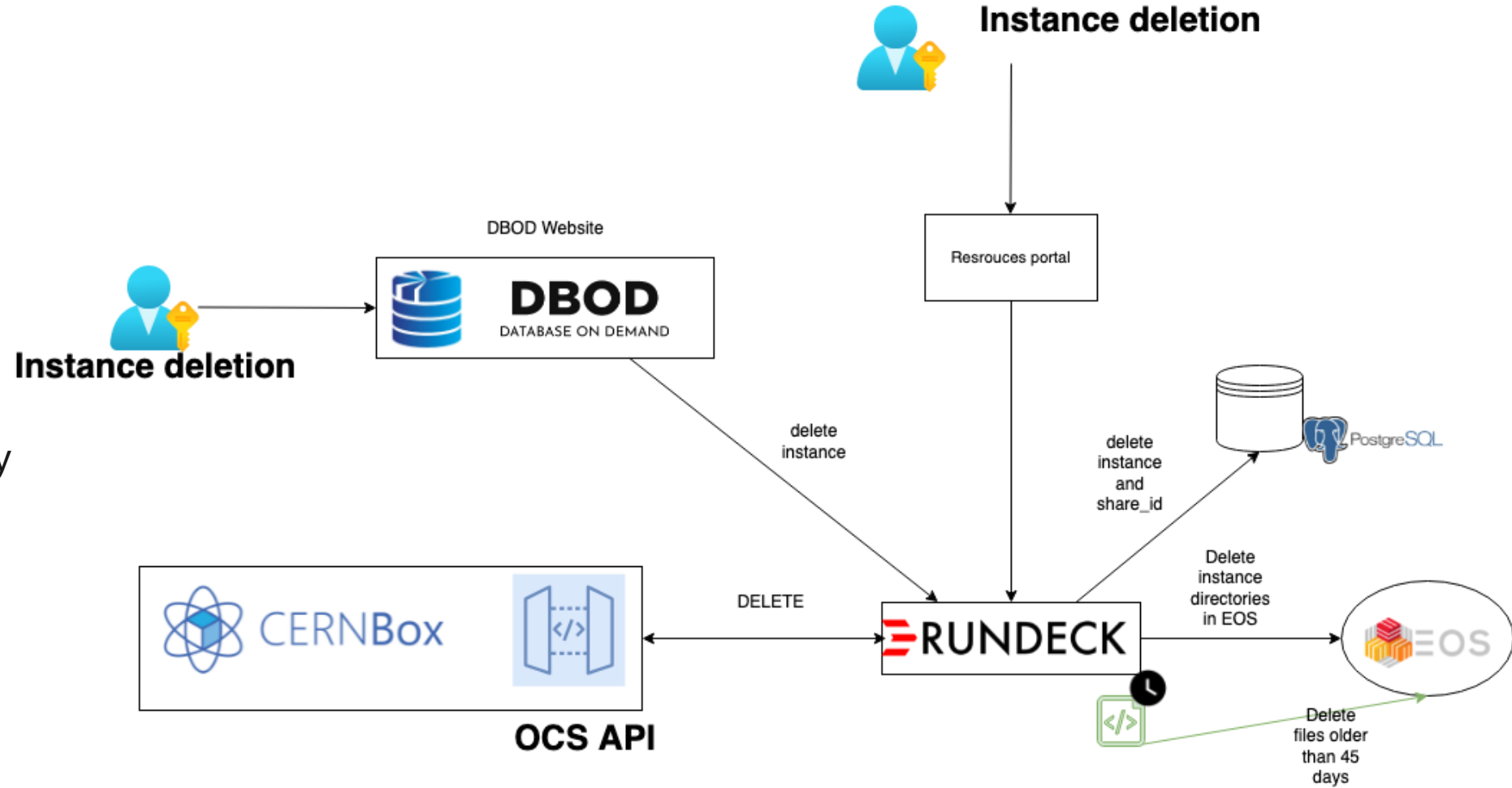
```
curl -X DELETE  
'https://cernbox.cern.ch/cernbox/desktop/ocs/v1.php/apps/files_sharing/api/v1/shares/418053' -u <user>:<password> -i
```

- Update a share:

```
curl -X PUT  
'https://cernbox.cern.ch/cernbox/desktop/ocs/v1.php/apps/files_sharing/api/v1/shares/418053?format=json' --data  
'shareType=0&shareWith=username&permissions=1&path=%2F__myprojects%2Fdbod%2Fpublic%2Finstance_name' -u mysql -i
```

Lifecycle of shares and files: deletion

- Daily:
 - Delete files older than 45 days
- On instance deletion:
 - Remove CERNBox share
 - Remove EOS directory



Results

- **Around 800 GB of data stored**
- **Fully automated process for sharing files with the instance owners**
- **Secure and easy way of sharing files with instance owners**
- **Reusable directories for sharing other types of files with the instance owners**

Road Map

- **As there is no way of knowing if the user specified an incorrect egroup (Sharing with non existent egroups is “possible”) -> validate egroup existence**
- **Provide support for instances in the TN -> Cannot access EOS with XRootD protocol from the TN-> FUSE mounts suggested**



**Thank you for your attention.
Questions?**