INDIGO-DataCloud Tutorial - Onedata Solutions for the LBT Distributed Archive

This tutorial will show you how to manage an astronomical distributed archive using INDIGO-DataCloud solutions.

In particular, you will learn how you can use Onedata to access, store and publish data using Cloud solutions for archiving astronomical files.

As system requirements you need at least 15 GB of free space and a software to run virtual machines (suggested VirtualBox).

This tutorial is divided in two steps. First you will learn how to deploy Oneprovider and Onezone. Second you will learn how you can use some Onedata features to manage a distributed astronomical archive.

Before going to Step 1, please have a look at the next 3 sub-sections to learn something more about INDIGO-DataCloud, Onedata and LBT.

What about INDIGO-DataCloud

INDIGO - DataCloud develops an open source data and computing platform targeted at scientific communities, deployable on multiple hardware and provisioned over hybrid, private or public, e-infrastructures. By filling existing gaps in PaaS and SaaS levels, INDIGO-DataCloud will help developers, resources providers, e-infrastructures and scientific communities to overcome current challenges in the Cloud computing, storage and network areas. INDIGO-DataCloud receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement RIA 653549.

More information about INDIGO-DataCloud can be found at https://www.indigo-datacloud.eu.

What about Onedata

Onedata is a global data management system, providing easy access to distributed storage resources, supporting wide range of use cases from personal data management to data-intensive scientific computations.

The most important concepts of the platform to understand at the beginning include:

- **Spaces** - distributed virtual volumes, where users can organize their data
- **Providers** - entities who support user spaces with actual storage resources exposed via Oneprovider services
- **Zones** - federations of providers, which enable creation of closed or interconnected communities, managed by Onezone services.

More information about Onedata can be found in the Official Onedata documentation.

LBT Distributed Archive Use Case

The Large Binocular Telescope (LBT) is an astronomical optical telescope located on Mount Graham in southeastern Arizona. The LBT is an international collaboration of several partners, among which Italy (INAF: Istituto Nazionale di Astrofisica) and Germany (LBTB: LBT Beteiligungsgesellschaft).

According to partner argreement, data of the INAF partner must be stored in Trieste, while data of the LBTB partner in Heidelberg. This results in a distributed archive, with geographically separated storages.

The flexibility of Onedata can provide storage solutions suitable for LBT requirements. We just need two spaces, a INAF space supported by a Oneprovider in Trieste and a LBTB space supporter by a Oneprovider in Heidelberg.

In this tutorial we are not going to create a fully operative distributed archive, but it is just a collection of examples to mimic the real configuration in order to show the main Onedata features to create user, upload data and metadata, manage access and everything useful to create a distributed archive.

Step 1 - Deploy Onedata

To complete this tutorial you need to have a running instance of Onedata.

We have prepared a Virtual Machine with Onezone and Oneprovider already installed. You can download it at this URL https://owncloud.ia2.inaf.it/index.php/s/s9rIR4eozV2QR6H. Then follow the instructions in the Deploy Onedata with preconfigured VM section. This is the **recommended** way to complete this tutorial.

If you prefer to configure your own virtual machines, you can instead follow the instructions in the Deploy Onedata manually section.

Instead, if you want to start a new Onedata installation, the Onedata Team provides a getting-started GitHub repository to deploy Onedata components. In the official getting-started repository you can find several scenarios, in order to try all available Onedata configuration and deploy Onedata with all its features.

Deploy Onedata with pre-configured VM

In this section we are going to deploy Onedata using a pre-configured Virtual Machine (VM). If do not want to use the pre-configured VM, but you prefer to configure your own virtual machines, jump to the next section Deploy Onedata manually. This method is derived from the scenario 2.0 of the official Onedata getting started guide.

Download the VM at this URL https://owncloud.ia2.inaf.it/index.php/s/s9rIR4eozV2QR6H. Import the `.ova` file in VirtualBox and run the VM.

WARNING: This error can occur related to your VirtualBox network interface configuration

```
Error: "Could not start the machine ... because the following physical network interfaces were not found:
vboxnet0 (adapter 1)
You can either change the machine's network settings or stop the machine."
```

You can easily fix it adding a new *Host-only network adapter* following this instructions http://islandora.ca/content/fixing-missing-vboxnet0.

The login credentials for the VM are

```
Username: lbt
Password: s3cret

Username: root
Password: s3cret
```

The present GIT repository is already cloned in the home of the `lbt` user, and the `odlbt` package is already installed. However, be sure to have the last version, running

```
cd /home/lbt/indigo-lbtda-onedata
git pull
cd odlbt
sudo make install
```

Onedata components are deployed using docker containers. If you are intersted in docker, check the website https://www.docker.com. You do not need to be a docker expert to complete this tutorial. Just imagine that a docker container contains everything required to make a piece of software (Onedata in our case) running.

As root, check if docker is running

```
service docker status
```

if it is not running, start it

```
service docker start
```

To deploy Onezone and the two Oneproviders you need three different root shells. Run the three following commands, one per shell.

```
run_onedata scenario2.0 zone
run_onedata scenario2.0 provider Trieste
run_onedata scenario2.0 provider Heidelberg
```

In each shell wait for the final congratulation message. In particular wait Onezone is deployed before deploying the two Oneproviders. Each of these commands deploys a docker container, one for Onezone and two for the two providers. Each docker container has its how IP address as listed in the Table 1.

Note that only `root` can deploy Onezone and Oneprovider, while you should run the other commands of this tutorial as `lbt` user.

docker container	Name	IP in VM
1	LBT Zone	172.18.0.2
2	Trieste	172.18.0.3
3	Heidelberg	172.18.0.4

Table 1: IPs of the Onedata installation with pre-configured VM

Congratulation, now Onedata is running! You can now jump to Step 2 to continue the tutorial. We will assume that all the next commands are executend inside the VM.

If you want to clean up all your previous attempts, run as lbt user (not as root)

```
clean_onedata
```

all configuration files will be deleted, allowing you to restart this tutorial from the beginning.

Remember that the access URL of your LBT Zone webpage is `https://172.18.0.2`.

Note for KVM users If you use qemu/kvm you may need to convert the `.ova` file to a format appropriate for qemu/kvm. You can check this link if you need help.

Deploy Onedata manually

In this section we are going to deploy Onedata in three different virtual machines. This is only for expert Onedata users, so if you do not feal confident, go back to the previous section Deploy Onedata with preconfigured VM. Instead, if have already downloaded and run the pre-configured Virtual Machine, skip this section and jump to Step 2 to continue the tutorial. This method is derived from the scenario 3.0 of the official Onedata getting started guide.

First of all create three virtual machines in your virtualization infrastructure. We suggest they have at least 2GB of RAM, and 16GB of local storage. We assume that the IP addresses or hostnames are those listed in Table 2. If your IP addresses or hostnames are different, remember to change them later in the `.yml` files.

hostname	Name	IP
onezone	LBT Zone	192.168.56.50
oneprovider1	Trieste	192.168.56.61
oneprovider2	Heidelberg	192.168.56.62

Table 2: IPs of the Onedata installation with three different machines

Warning. Onedata uses these ports, therefore check your firewall:

- 7443
- 6666
- 8876
- 5556
- 9443
- 8877
- 6665
- 5555
- 80
- 8443
- 443
- 53

Then follow these steps for all the 3 machines.

1. Install CentOS 7. It is not mandatory, but warmly suggested to use CentOS. Be sure your system is up to date

   ```
   yum update
   yum install wget
   ```

2. Check hostname, IP address and ports. Be sure that each machine can see to the two others.
3. Install docker and docker-compose

   ```
   tee /etc/yum.repos.d/docker.repo <<-'EOF'
   [dockerrepo]
   name=Docker Repository
   baseurl=https://yum.dockerproject.org/repo/main/centos/7/
   enabled=1
   gpgkey=https://yum.dockerproject.org/gpg
   gpgcheck=1
   EOF

   curl -L https://github.com/docker/compose/releases/download/1.8.0/docker-compose-`uname -s`-`uname -m`
   chmod +x /usr/local/bin/docker-compose

   yum install docker-engine
   ```

4. Install GIT

   ```
   yum install git
   ```

5. Install some python requirements

   ```
   curl "https://bootstrap.pypa.io/get-pip.py" -o "get-pip.py"
   python get-pip.py
   pip install docopt
   pip install numpy
   pip install pyfits
   ```

6. Clone the GIT repository (we suggest to clone the repository as normal user, not as root)

```
git clone https://github.com/turambar173/indigo-lbtda-onedata.git
```

7. Edit the `.yml` files with the IP of your machines. Enter the directory

   ```
   cd indigo-lbtda-onedata/odlbt/scenarios
   ```

   Edit the file scenario3_0_Heidelberg/docker-compose-oneprovider.yml at the lines 62 and 66 entering respectively the IP of the machine hosting Heidelberg provider and the IP of the machine hosting Onezone. In the same way edit the file scenario3_0_Trieste/docker-compose-oneprovider.yml at the lines 62 and 66 entering respectively the IP of the machine hosting Trieste provider and the IP of the machine hosting Onezone. The `.yml` file of the Onezone does not need to be changed.

8. Install the `odlbt` package

   ```
   cd indigo-lbtda-onedata/odlbt
   sudo make install
   ```

9. Install oneclient (if you need help check this link)

   ```
   curl -sS  http://get.onedata.org/oneclient.sh | bash
   ```

10. Remember to start the docker daemon

    ```
    service docker start
    ```

To deploy Onezone and the two Oneproviders you need to run as root the three following commands, the first in the Onezone machine, the second in the Trieste Oneprovider machine and the third in the Heildeberg Oneprovider machine.

```
run_onedata scenario3.0 zone
run_onedata scenario3.0 provider Trieste
run_onedata scenario3.0 provider Heidelberg
```

In each shell wait for the final congratulation message. In particular wait Onezone is deployed before deploying the two Oneproviders. Each of these commands deploys a docker container, one for Onezone and two for the two providers.

Note that only `root` can deploy Onezone and Oneprovider, while you should run the other commands of this tutorial as normal user.

Congratulation, now Onedata is running! You can now jump to Step 2 to continue the tutorial.

If you want to clean up all your previous attempts, run as lbt user (not as root)

```
clean_onedata
```

all configuration files will be deleted, allowing you to restart this tutorial from the beginning.

Remember that the access URL of your LBT Zone webpage is `https://<onezone-ip>`.

Step 2 - Use Onedata

Now we are going to use the Onedata instance you just deployed. We assume that all the shell commands are executed as normal user, not root.

You can access Onedata using a web browser (hint: use Firefox), therefore go to your LBT Zone webpage (the URL is https://172.18.0.2 if you are using the pre-configured VM) and sign in. Onedata comes with the admin user already created with these credentials

```
Username: admin
Password: password
```

You can access Onedata also through REST API and CDMI API. These methods are more flexible and suitable for our purposes. In addition, we have created the `odlbt` package to avoid the user to remember long and cryptic GET and POST requests.

The `odlbt` package is already installed in the VM. Conversely, if you have to install it, enter the odlbt directory inside this GIT repository and run

```
cd indigo-lbtda-onedata/odlbt
sudo make install
```

The package `odlbt` basically works running commands in this way

```
odlbt <command> [options]
```

You can have help running

```
odlbt --help
odlbt <command> --help
```

Configure odlbt

Before using it, we need to configure the `odlbt` package, setting up IPs of our Onedata installation and user credentials. The configuration file is normally stored in `~/.odlbtconfig`.

First create a json file for the admin credentials and save it as `admin.json`

```
{
    "username" : "admin",
    "password" : "password",
    "userRole" : "admin"
}
```

Then add the admin credentials to the configuration file

```
odlbt config --global user admin.json
```

You can now delete the `admin.json` file.

Now we are going to add to the configuration file the IPs of Onezone and Oneproviders

```
odlbt config --global zone <onezone-ip>
odlbt config --global provider Trieste <trieste-provider-ip>
odlbt config --global provider Heidelberg <heidelberg-provider-ip>
```

If you are using the pre-configured VM, use the IPs listed in the Table 1 for the Trieste and Heidelberg provider, otherwise use the IPs of your virtual machines created in Deploy Onedata manually section.

Now create an access token for the admin user, and store it in the configuration file

```
odlbt config --global token admin
```

The access token is necessary to use CDMI API. To have a look of your token and to check if the token has been correctly create run

```
odlbt user get token admin
```

if an empty string or `null` is returned, something went wrong with token creation, therefore run the token creation command again.

To create the user token we have used the client tokens REST API. If you are interested in more information about it, check this link.

Create users

Now we are going to create a new user with the `odlbt user` command, using REST API.

First, prepare the json file `newuser.json` with the user credential. In this case we have to set `userRole` to `regular`.

```
{
    "username": "newuser",
    "password": "password",
    "userRole": "regular"
}
```

Then create the user

```
odlbt user add newuser.json
```

Then add the newly create user to config

```
odlbt config --global user newuser.json
```

You can now delete the `newuser.json` file.

Finally, go to LBT Zone webpage and sign in with the `newuser` credentials.

For more information about the used REST API check this link.

Create spaces and get support

Now we are going to create new spaces with the `odlbt space` command, using REST API.

Create the `INAF` space running

```
odlbt space add INAF
```

If you now access to the LBT Zone webpage as `admin`, you will see the newly created space in the left panel. But you can not yet upload files, you need to get support from a provider. Let's get some support!

```
odlbt space support INAF Trieste 1024
```

With this command we support the INAF space with the Trieste provider with 1024 MB.

Just to let you know, you can ask support from multiple providers. In this case Onedata will manage how uploaded files are distributed among the providers.

To complete the space configuration for the LBT distributed archive, create a space for LBTB and get support from Heidelberg Oneprovider

```
odlbt space add LBTB
odlbt space support LBTB Heidelberg 1024
```

For the LBT distributed archive use case we just need the two spaces INAF and LBTB, but if you want more and a complex configuration you can do it. If you are interested, please check the Onedata documentation webpage.

For more information about the used API check these links to create spaces, create space tokens, get space tokens and support space.

Moreover, you can also create and support space using the webpage, in this case you need to get the space token from the webpage itself or running

```
odlbt space get INAF token
```

More informations about this topic are available in the official Onedata documentation at this link and at this link.

Upload data

Now it is time to upload some data to your spaces. You can download some public LBT files at this URL https://owncloud.ia2.inaf.it/index.php/s/mV50adXpRFXaqWR. Untar the archive and use the files in the folder *Test files* to perform some upload test.

The easiest way to upload data is using the web interface. Go to your LBT Zone webpage and sign in as admin. Now you can see in the map of the world two green icons: these are the providers that support your spaces. Click on the Trieste provider icon on the map and click on the button *Go to your files*. You will be redirected to Oneprovider web interface, where you can access and manage your files.

To upload a file, simply drag and drop the file into the browser window, or use the button *Upload file*.

Once in the Oneprovider interface with the tab menu on the left, you can easily:

- switch between your spaces in the Spaces tab;
- access and manage data in Data tab;
- create and manage user groups in the Groups tab.

Note that you can explore all your spaces, even if they are not supported by the Oneprovider you are now connected to.

More informations about this topic are available in the official User Quickstart Guide, Space Management Page and File Management Page.

Access your spaces with POSIX

In Onedata, files can also be accessed directly via POSIX protocol, using Oneclient tool. With Oneclient you can mount your spaces in your local file system tree.

The basic command line syntax to mount spaces using a specific Oneprovider is

```
oneclient -i -H <PROVIDER_HOSTNAME> -t <ACCESS_TOKEN> <MOUNT_POINT>
```

So create the mount point, for instance

```
mkdir ~/MySpaces
```

Then retrive your access token

```
export ONECLIENT_ACCESS_TOKEN=$(odlbt user get token admin)
```

and set the provider IP, e.g. Trieste, as environment variable

```
export ONECLIENT_PROVIDER_HOST=172.18.0.3
```

Then you can mount your spaces just running

```
oneclient -i ~/MySpaces
```

Now we are going to upload some data of the two LBT partners, INAF and LBTB, using Oneclient. We will use the LBT-Public files downloaded previously. In the INAF folder there are some files of the INAF partner, while in the LBTB folder some files of the LBTB partner. You can check it by the PARTNER keyword in the fits headers.

First, enter and explore the spaces you just mounted

```
cd ~/MySpaces
```

create the folders with the year number

```
mkdir INAF/2015
mkdir INAF/2014
mkdir LBTB/2014
```

then upload/copy files of INAF in INAF space according to the year

```
cd ~/LBT-Public/INAF
cp -v luci.2014*fits ~/MySpaces/INAF/2014/
cp -v luci.2015*fits ~/MySpaces/INAF/2015/
```

and files of LBTB in LBTB space

```
cd ~/LBT-Public/LBTB
cp -v luci.2014*fits ~/MySpaces/LBTB/2014/
```

Now we have finished to upload data, so we can unmount the spaces

```
oneclient -u ~/MySpaces
```

You can check in the webpage that everything is in the right place. Go to your LBT Zone webpage and sign in as admin. Click on the Trieste (or Heidelberg if you prefer) provider icon on the map and click on the button *Go to your files*. You will be redirected to Oneprovider web interface, where you can explore your files. Note that you can access files in all your spaces, even those stored in the provider you do not select in the map.

More informations about this topic are available in the official Oneclient documentation.

Add metadata

For an astronomical archive metadata are crucial. They describe an astronomical resource (image, spectrum, calibration, etc.) for purposes such as discovery and identification.

Now INAF and LBTB data are in Onedata, so let's add metadata to them. Onedata uses metadata in json format and they can be uploaded with CDMI API.

In the `odlbt` package, the `extract-metadata` command gets some important metadata (instrument name, sky coordinates, exposure time, etc.) from the fits header, and saves them in a json file suitable to be loaded in Onedata. Generate the json metadata files for all the INAF files you have uploaded in the previous section

```
cd ~/LBT-Public/INAF
extract-metadata *fits
```

Now json files can be uploaded as metadata in Onedata with this command

```
odlbt metadata add -p <provider-name> -s <space-name> <file.json>...
```

where you have to set the Oneprovider name and the space name where the files are, including the path in Onedata. This command will connect to provider-name using CDMI API and upload file.json metadata to file stored in space-name. Therefore you will run something like

```
odlbt metadata add -p Trieste -s INAF/2014 luci.2014*.json
odlbt metadata add -p Trieste -s INAF/2015 luci.2015*.json
```

To check that the metadata have been correctly added enter the webpage, go to the INAF space and select the *Show metadata* button at the right of the file name.

Repeat the same procedure also for LBTB files.

In principle you can use any of your provider as provider-name, but it is better to use the provider where data you want to add metadata to are. Otherwise, if you are directly connected to a provider where data are not, Onedata will connect the first Oneprovider to the second Oneprovider to get the file, then add the metadata, and then sends back the metadata to second Oneprovider with a lack of efficiency.

So, for the LBTB files you shuold use the `-p Heildeberg` option running `odlbt metadata` command.

For more information about CDMI please refer to the Official Onedata CDMI documentation.

Manage access to data

Onedata allows to share data between users. This is done exchanging invitation token. To test this Onedata feature assume you want to allow the newuser created before to access INAF files of the year 2014.

Go to your LBT Zone webpage and sign in as admin. Click on the Trieste provider icon on the map and click on the button *Go to your files*. In the right tab menu select *Spaces*, to enter the space management screen. Click the button *Invite user*, in the top right. Onedata will show you a pop-up with the token to share with newuser. Copy the token in a text editor for later use.

Now logout as admin, and sign in as newuser. The newuser does not have any space, so in the world map there are no provider icons.

In the left menu select *Data space management*, then click the button *Join a space*. Finally, enter the token copied previuosly and click *Join* to join the INAF space.

Now in the world map, also newuser have Trieste Oneprovider and can access all data in the INAF space. But we want to limit access to 2014 data only.

So, logout newuser, sign in as admin, and *Go to your files* in the Trieste provider.

Select the 2015 folder (only one click), then select the *Change element permissions* button in the top menu (the one with the padlock icon). As Permission type select ACL, then *Add* a new permission rule. In the dropdown menu select newuser, and as Type select Deny. Then click OK to save the new permission rule.

If you now logout admin, and sign in as newuser again, you will see that the folder 2015 has disappeared from INAF space for newuser.

You can find more details about access control and element permissions in the Official Onedata documentation.

Conclusions

Congratulation, you have completed the INDIGO-DataCloud Tutorial about Onedata Solutions for the LBT Distributed Archive.

If you found any bug or issue, please report to bignamini@oats.inaf.it.

We really appreciate if you can provide us some feedback filling this questionnaire.

INDIGO-DataCloud receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement RIA 653549