Syntool → Installation

Installation guide

OceanDataLab | syntool@oceandatalab.com | 2018-10-12

Introduction

This document will guide you through the installation of the Syntool platform.

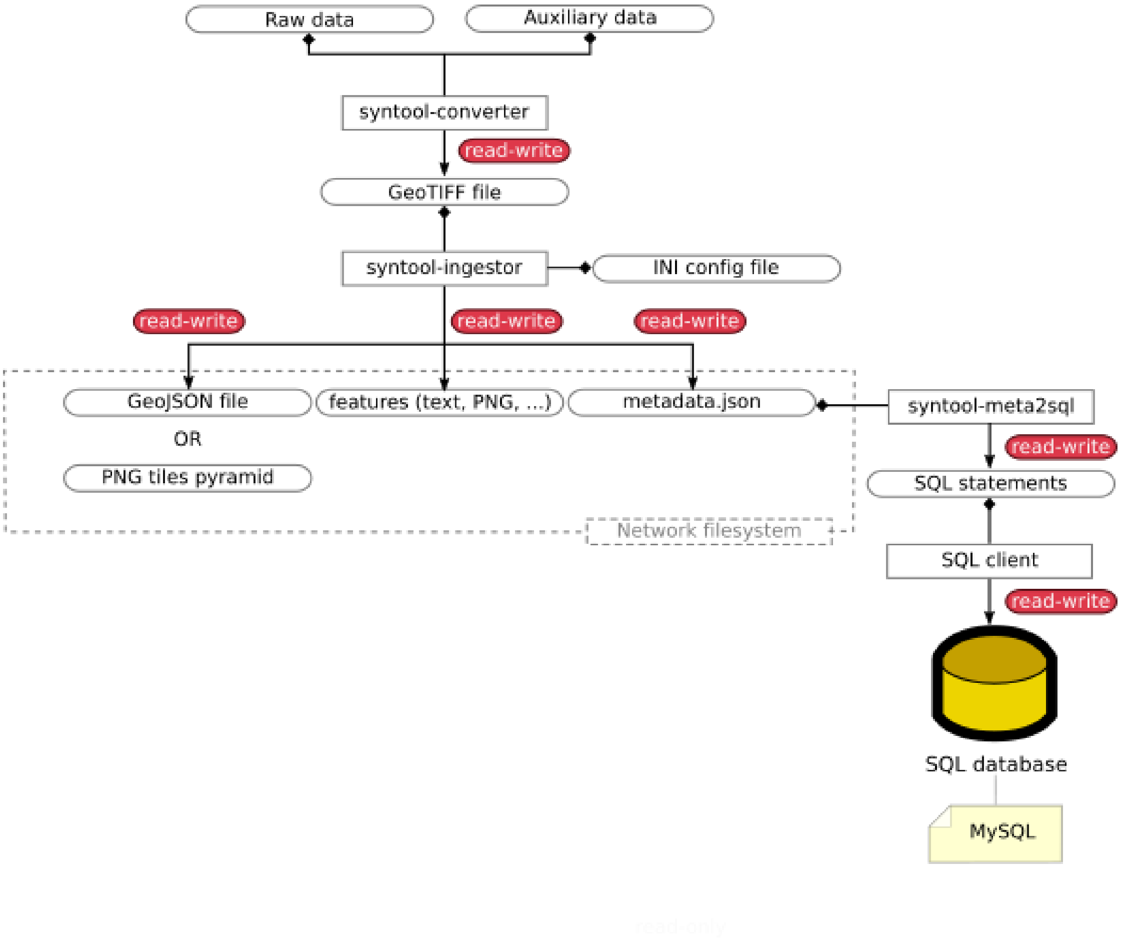
It is composed of two main sections: the first one provides instructions for the installation of the processing environment while the second section focuses on the deployment of the web components for Syntool.

While it is possible to install the whole platform on a single machine, this guide describes the deployment process on two machines as the processing and the web components have different requirements.

Processing machine

This machine will be used to transform raw data into Web-compatible representations that the Syntool web portal will be able to display. These web-compatible representations will be stored in the NFS share (or another distributed storage) so that the Web host can read them.

It will also extract temporal and spatial coverage from the raw files and store this information in a MySQL database. The Web host will exploit the content of this database to determine which data must be displayed, based on the viewport of the map and the state of the timeline in the Syntool portal.



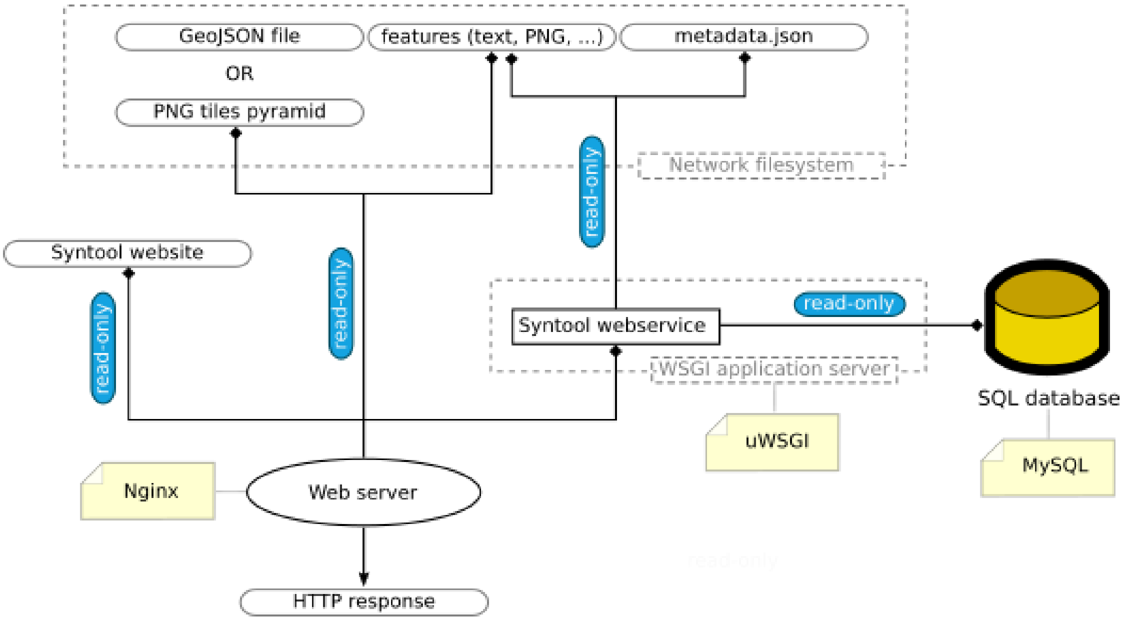
Some assumptions for the rest of this chapter:

- The system is a minimal installation of Ubuntu 18.04.
- The NFS share has been mounted as '/mnt/syntool' with the read-write option

In this installation manuel, the IP address for this machine will be 111.222.333.444, remember to adapt to replace it by the actual IP address.

Web machine

This machine will host the web services and the Syntool portal. The HTTP server (Nginx) will also give access to the content of the NFS share where the Web-compatible data representations are stored.



Some assumptions for the rest of this chapter:

- The system is a minimal installation of Ubuntu 18.04.
- The NFS share has been mounted as '/mnt/syntool' with the read-only option

In this installation manuel, the IP address for this machine will be 555.666.777.888, remember to adapt to replace it by the actual IP address.

Processing tools (backend)

Dependencies installation

Syntool relies on a large number of libraries for data loading, image processing, geographic projections, etc...

Most of them are also dependencies for the GDAL library, so installing GDAL (development version) using the system package manager covers almost all software requirements.

All the commands listed in this section must be executed as root or must be prefixed with the `sudo` command if the user has sufficient permissions.

```
apt install python-tk python-pip gdal-bin libgdal-dev libgrib-api-dev
```

> ### *Note*
>
> grib-api has been replaced by eccodes on recent systems, so if the installation of the pygrib Python package fails at a later stage of this guide due to the lack of eccodes, you can install it with:
>
> apt install libeccodes-dev

A MySQL client will also be required to export the temporal and spatial coverages to the database

```
apt install mariadb-client
```

HDF4 library issue

The name of the dynamic libraries for HDF4 (that were automatically installed as a GDAL dependency) ends with an "alt" suffix which causes issues with the Python wrapper for these libraries (python-hdf4). Adding symbolic links with the standard name of these libraries will prevent compilation failures and make the installation process easier.

```
ln -s libdfalt.so /usr/lib/libdf.so
ln -s libmfhdfalt.so /usr/lib/libmfhdf.so
```

Common Python packages

The installation process of some Python dependencies (gdal, cerinterp and ceraux) will fail if Cython, Numpy and pyproj are not already available [1] so it is mandatory to install these three packages first.

It is recommended to install the Python packages in a virtual environment (see [4] to avoid version conflicts and issues during system upgrades. You can also use the `--user` switch of the `pip` command to install packages in your home directory. In both cases, none of the following commands should require root account or sudo permissions unless specified.

```
pip install cython
pip install numpy
pip install pyproj
```

In case there is an error during the installation of pyproj, you can try to install it directly from the Github repository:

```
apt install git  # root or sudo
pip install git+https://github.com/jswhit/pyproj.git
```

The installation of the Python package for GDAL is not straightforward, it must be customized to match (or be close to) the version of the GDAL C library installed on the system.

```
gdal_version="$(gdal-config --version)"
echo "${gdal_version}"
```

Versions 2.3.0 and above of the GDAL Python package are not compatible with Syntool at the moment, so you may have to edit the `pip download` command in the following script

```
workspace="$(mktemp -d)"
cd "${workspace}"

# if gdal_version is equal or above 2.3.0, replace the following line by
# pip download "gdal<2.3.0"
pip download "gdal<=${gdal_version}"

# The version of GDAL downloaded by pip depends on the content of the Pypi
# repository at the time you ran the "pip download" command, so this script
# uses "find" commands instead of the actual name of the downloaded .tar.gz
# file and extracted directory
find . -name "*.tar.gz" -exec tar xzf {} \;
cd "$(find . -mindepth 1 -maxdepth 1 -type d)"

python setup.py build_ext --include-dirs=/usr/include/gdal/
python setup.py install

# Cleanup
cd
rm -rf "${workspace}"
```

Cerbere

Cerbere is developed by the LOPS (Laboratoire d'Océanographie Physique et Spatiale) at IFREMER. The tools it provides help with reading and manipulating geophysical data and are used by some Syntool data readers.

```
pip install https://ftp.odl.bzh/odl/syntool/libs/cerform-0.1.0.tar.gz
pip install https://ftp.odl.bzh/odl/syntool/libs/cerinterp-0.1.0.tar.gz
pip install https://ftp.odl.bzh/odl/syntool/libs/ceraux-0.1.0.tar.gz
pip install https://ftp.odl.bzh/odl/syntool/libs/cerbere-0.1.738.tar.gz
```

Syntool processing tools

```
pip install https://ftp.odl.bzh/odl/syntool/libs/sar4ocean-0.0.71.tar.gz

pip install https://ftp.odl.bzh/odl/syntool/syntool_converter-0.1.614.tar.gz

# pip will probably issue a warning about conflicting versions of GDAL during
# the installation of the syntool_ingestor package. You can ignore it safely.
pip install https://ftp.odl.bzh/odl/syntool/syntool_ingestor-0.1.300.tar.gz

pip install https://ftp.odl.bzh/odl/syntool/syntool_exporter-0.1.14.tar.gz
```

Configuration and auxiliary files

The syntool-ingestor tool requires a configuration file to define the projection and viewport of the Syntool portal in order to properly generate the images.

Command line arguments passed to syntool-ingestor can be quite verbose, but they can also be read from a file.

Some auxiliary data may also be required by syntool-converter readers, such as lookup tables for Sentinel-3 atmospheric correction.

The following commands will allow you to fetch samples for these files. The archive can be extracted anywhere, the `/etc/syntool` directory is just an example (which will require root/sudo permissions):

```
mkdir -p /etc/syntool

wget https://ftp.odl.bzh/odl/syntool/syntool_config_samples.tar.gz \
     -O /tmp/syntool_config_samples.tar.gz
tar xzf /tmp/syntool_config_samples.tar.gz -C /etc/syntool/

wget https://ftp.odl.bzh/odl/syntool/syntool_aux_samples.tar.gz \
     -O /tmp/syntool_aux_samples.tar.gz
tar xzf /tmp/syntool_aux_samples.tar.gz -C /etc/syntool/
```

Side notes

PATH update

You may have to execute the `hash` or `rehash` [2] commands for the Syntool commands to be recognized.

Broken pip install

On some systems the `pip install` command does not work anymore after following this installation procedure. It is due to an incompatibility between the precompiled `pyopenssl` Python package and the OpenSSL dynamic libraries installed on the system [3].

To fix it:

```
apt install libffi-dev libssl-dev
pip uninstall -y pyopenssl
pip install --no-binary :all: pyopenssl
```

Connection to MySQL

Credentials are required for the connection to the MySQL database at the end of the processing chain (to register the temporal and spatial coverage).

In order to pass these credentials to the MySQL server without including them in the command line (which is bad practice for security reason), you can create a configuration file named `.my.cnf` in your `${HOME}` directory:

```
# Edit the user and password to match your actual credentials
cat > ~/.my.cnf <<EOF
[client]
user=syntool_processor
password=REDACTED2
EOF
```

Sentinel-1 raw data

Sentinel-1 data are usually available as as compressed "SAFE" directories: it is necessary to decompress them before Syntool can read and convert them into GeoTIFF files. To do so, the `unzip` program must be installed:

```
apt install unzip
```

Done!

The installation of the processing tools for Syntool is now complete.

You should check the help message for each command (and consult documentation when it is available).

```
syntool-converter --help
syntool-colorbar --help
syntool-ingestor --help
syntool-meta2sql --help
```

Web portal (frontend)

Dependencies installation

The web machine must host an HTTP server (Nginx) to expose Syntool resources (website, PNG tiles, GeoJSON, etc...) on the Internet.

A Python applications server (uWSGI) must also be installed to handle the WSGI webservice which queries the database to reply to search requests sent by the Syntool portal.

All the commands listed in this section must be executed as root or must be prefixed with the `sudo` command if the user has sufficient permissions.

```
apt install python-pip
apt install libmysqlclient-dev
apt install nginx
```

MySQL server

A MySQL server can be installed and configured safely with the following commands:

```
apt install mysql-server
mysql_secure_installation
```

The default configuration states that the MySQL server only listens on the loopback device, so the configuration must be altered so that MySQL listens on the local network interface that can be reached by the processing machine.

```
# Replace 111.222.333.444 by the actual IP address of the processing machine
sed -i 's/^bind-address\s*=.*/bind-address=111.222.333.444/' \
    /etc/mysql/mysql.conf.d/mysqld.cnf
systemctl restart mysql

# Make sure MySQL will be started automatically after a reboot
systemctl enable mysql
```

Once MySQL is properly configured, create a database and change its charset to avoid issues when importing the output of the `syntool-meta2sql` command (which produces SQL statements on the processing machine).

```
mysql -e 'CREATE DATABASE syntool'
mysql -e 'ALTER DATABASE syntool CHARACTER SET latin1 \
                                 COLLATE latin1_general_ci;'
```

Finally, create two users and set their permissions on the tables of the syntool database:

- the syntool_webservice user (connecting from this machine) will only be allowed to read data
- the syntool_processor user (connecting from the processing machine, i.e. 111.222.333.444) must have the permission to create tables, insert data, etc...

```
mysql -e "GRANT SELECT, LOCK TABLES, SHOW VIEW, EVENT, TRIGGER ON syntool.* \
          TO 'syntool_webservice'@'localhost' IDENTIFIED BY 'REDACTED1';"

# Replace 111.222.333.444 by the actual address of the processing machine
mysql -e "GRANT ALL PRIVILEGES ON syntool.* \
          TO 'syntool_processor'@'111.222.333.444'  IDENTIFIED BY 'REDACTED2';"
```

Webservice

MySQL initialization

Mandatory SQL tables must be created in the database for the webservice to work properly. The `syntool.sql` file contains the SQL statements that will create these tables. You can fetch and execute these statements with the following commands:

```
wget https://ftp.odl.bzh/odl/syntool/syntool.sql -O /tmp/syntool.sql
mysql syntool < /tmp/syntool.sql
```

The `syntool.sql` file can also be generated on the processing machine (where the `syntool_exporter` package is installed):

```
syntool-meta2sql --init-only syntool.sql
```

Python package & WSGI server

The Python package for the webservice and the Python applications server can be installed with the `pip` command.

```
pip install https://ftp.odl.bzh/odl/syntool/syntool_metadata-0.1.106.tar.gz
pip install uwsgi
```

Configuration

The webservice and the Python applications server must be configured using files in INI format. The following commands will allow you to download samples for these configuration files and save them in the `/etc/syntool` directory.

```
mkdir -p /etc/syntool

# Sample configuration for the webservice
wget https://ftp.odl.bzh/odl/syntool/syntool.ini \
  -O /etc/syntool/syntool.ini

# Sample configuration for the Python applications server
wget https://ftp.odl.bzh/odl/syntool/uwsgi.ini \
  -O /etc/syntool/uwsgi.ini
```

The `/etc/syntool/syntool.ini` file must be edited to provide the correct password for the `syntool_webservice` MySQL user.

Once the password is set, start the webservice manually with the following command:

```
uwsgi --ini /etc/syntool/uwsgi.ini
```

Quit with Ctrl+C and check that there is no error message in the `/var/log/syntool-uwsgi.log` file

Starting the service automatically

Most Linux distribution now rely on `systemd` and unit files to manage daemons. A unit file for the Syntool webservice is available on the ODL website and must be deployed before `systemd` can use it.

```
wget https://ftp.odl.bzh/odl/syntool/syntool.service \
     -O /etc/systemd/system/syntool.service

systemctl start syntool.service
```

If the webservice is properly configured and no error message appears, you can enable the service to tell systemd that the Syntool webservice must be started automatically.

```
# Make sure the webservice will be started automatically after a reboot
systemctl enable syntool.service
```

> ***Warning***
>
> Remember to set a logrotate mecanism to manage the size of the log files!

Website

The Syntool websites are generated using templates and configuration files. The following commands will allow you to create an empty portal which is required to test the success of the deployment process when you reach the end of this guide.

```
# Download source code for the website generator
wget https://ftp.odl.bzh/odl/syntool/syntool_webapp.tar.gz \
     -O /tmp/syntool_webapp.tar.gz
cd ~
tar xzf syntool_webapp.tar.gz
cd syntool

mkdir -p /srv/http

# Prepare the web resources
./developer/generate.sh --prepare

# Deploy them
./developer/generate.sh --package /srv/http/syntool

# Generate the web page for the portal using a sample configuration
./developer/generate.sh \
  --portal ./developer/portal /srv/http/syntool/index.html "Syntool portal"
```

When some data tiles are not found, OpenLayers (the library used by Syntool for rendering data in a map) displays a pink image instead. Using a transparent image as a fallback provides a much better looking result. Download the transparent image and save it in the Syntool website directory:

```
wget https://ftp.odl.bzh/odl/syntool/pixel.png \
     -O /srv/http/syntool/pixel.png
```

Create a symbolic link named `data` that points to the directory of the NFS share where the files produced on the processing machine are stored.

```
ln -s /mnt/syntool/ /srv/http/syntool/data
```

HTTP server

A configuration file for Nginx is available on the ODL website, download it and put it in the directory dedicated to Nginx virtual sites:

```
wget https://ftp.odl.bzh/odl/syntool/nginx.conf \
     -O /etc/nginx/sites-available/syntool.conf
```

Nginx provides a virtual site that is configured to be the default target for HTTP requests. The Syntool website should now be the default target, so remove the default virtual server and activate Syntool instead.

```
# The website-nginx configuration states that it should be used as the
# default website, so remove the existing default
rm /etc/nginx/sites-enabled/default

# Activate the Syntool website
ln -s /etc/nginx/sites-available/syntool.conf \
      /etc/nginx/sites-enabled/syntool.conf
```

Check that everything is in order and restart Nginx for the changes to take effect.

```
# Check that there is no syntax error in the configuration files
nginx -t

systemctl restart nginx
```

Opening http://555.666.777.888/ (replace by the actual IP address for the frontend machine) in a web browser should now display the Syntool portal.

Enable the Nginx unit file so that the HTTP server is started automatically too.

```
# Make sure Nginx will be started automatically after a reboot
systemctl enable nginx
```

Done!

The Syntool portal is now installed, you can tweak the configuration files in `~/syntool/developer/portal` and update the portal by repeating the commands that involve the `~/syntool/developer/generate.sh` script (see Website).

Please read the Syntool webapp documentation to learn how the configuration files work.

Questions, feature requests and bugs

Please register and post your questions, feature requests and bug reports on the Syntool forum: https://forum.oceandatalab.com/forum-5.html

[1] Their setup.py file either imports cython/numpy/pyproj or links to the dynamic libraries provided by Cython, Numpy or pyproj.

[2] The command to refresh the list of executables found in the PATH depends on your shell configuration.

[3] Always test that packages work as intended after the installation! By default the pip command installs precompiled versions of the packages to reduce the requirements to install them (compiler, header files, etc...) and simplify the installation process (just copy the files in the right directory).

Simpler and faster, but less reliable: you have to check that your system is compatible with the libraries used for the compilation of the binaries, i.e. the precompiled binaries will not work if the dynamic libraries installed on your system lack symbols that were available on the packager's machine.

[4] https://virtualenv.pypa.io/en/stable/