Installing and configuring CLARIN-DSPACE 7

Author: Michele Mallia

Supervisor: Riccardo Del Gratta

mail: michele.mallia@ilc.cnr.it | michele.mallia@cnr.it | riccardo.delgratta@ilc.cnr.it

Pisa : Gennaio 2026

Summary

1. Introduction 1
2. System information and requirements 2
3. Installation 4
   3.1 Installing Backend Requirements 4
      3.1.1 Installing Oracle Java JDK (17.0.7) 4
      3.1.2 Installing Apache Maven (3.8.1) 6
      3.1.2 Installing Apache Ant (1.10.15) 8
      3.1.3 Installing PostgreSQL (17) 9
      3.1.4 Installing Solr (8.11) 11
      3.1.5 Installing Tomcat (8.11) 11
      3.1.6 Installing Node JS (16) 16
      3.1.7 Installing Git 16
      3.1.8 Installing Yarn 17
      3.1.9 Installing LetsEncrypt 17
      3.1.10 Installing Shibboleth 17
   3.2 Installing DSPACE (backend + frontend) 22
      3.2.1 Create a DSPACE system user 22
      3.2.2 Downloading the DSPACE Sources 22
      3.2.3 Database Setup 23
      3.2.4 Create DSPACE folder 23
      3.2.5 Building the DSPACE Sources 24
      3.2.6 Copy Solr Cores 25
      3.2.7 Check database connection 25
      3.2.8 Create admin 26
      3.2.9 Deploy the Server webapp 26
      3.2.10 Deploy the Frontend sources 27
4. Configuration 28
   4.1 Generate LetsEncrypt Certificate 28
   4.2 Configure LetsEncrypt Certificate on Apache 29
   4.3 Create new configuration for Apache 31
   4.4 Generate OpenSSL Certificate for Shibboleth 32
   4.5 Configure etc/hosts file 33
   4.6 DSPACE Backend configuration 33
      4.6.1 Change log level of DSPACE 34

4.7 DSPACE Frontend configuration 35
4.7.1 Customization of graphic elements 37
4.8 Test connection between backend and frontend 49
4.9 Configure Shibboleth 50
5. Test 55
5.1 Apache 55
5.2 Tomcat 56
5.3 Shibboleth 56
5.4 DSPACE Backend 60
5.5 DSPACE Frontend 60
6. Migration procedure 61
6.1 DSPACE 5 Database Backup 61
6.2 Stopping Shibboleth, Apache, Tomcat, and DSpace Frontend Services 62
6.2.1 Stopping Shibboleth Service 62
6.2.2 Stopping Apache and Tomcat 62
6.2.3 Stopping the DSpace Frontend (Angular UI) 63
6.3 Deleting Old DSpace 7 Databases 63
6.4 Importing the New DSpace 7 Databases 64
6.5 Downloading the DSpace Migration Tool 65
6.6 Disabling Direct Download from S3 67
7. Start Handle Server 68
8. Change item submission 71
9. Change default license 73
10. Add Matomo JS Script for statistics 74

1. Introduction

This document provides a detailed description of the procedures for installing the new DSPACE-CLARIN[1] software (specifically the branch clarin v-7)[2] for managing scientific materials and updating data from version 5.x to version 7.x. There is a Dockerized version of the software, but it is only functional in a development and testing context; for the production version, a complete local installation procedure must be carried out.

Currently, the system is based on Ubuntu 24.04 LTS and the machine is hosted within our research institute (ILC – Institute of Computational Linguistics "Antonio Zampolli," located in the CNR Research Area of Pisa); it can be accessed either via VPN or through ports 80 and 443 (if an Apache server is configured to listen on those ports).

There are two main reference points for obtaining information about the installation procedures:

1. the first is located within the clarin-dspace repository (there is an internal wiki for v7)[3], which contains simplified procedures;
2. the second is in the software wiki (for the version 7.x)[4], which includes all the complete procedures and customizations.

We will proceed step by step, first outlining the specific characteristics of the current system and then detailing the steps required to set up a fully functional working environment.

[1] https://github.com/ufal/clarin-dspace

[2] https://github.com/ufal/clarin-dspace/tree/clarin-v7

[3] https://github.com/ufal/clarin-dspace/wiki/NewInstallation

[4] https://wiki.lyrasis.org/display/DSDOC7x/Installing+DSpace

2. System information and requirements

Here is a detailed overview of the specifications of the machine we will be working on:

- **Hostname**: dev-clarin
- **Operating System**: Ubuntu 24.04.02 LTS
- **Chip**: Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz
- **Total Number of Cores**: 2 core (2 total threads)
- **Kernel Version:** Linux 6.8.0-62-generic
- **Storage Capacity:** 310 GB
- **Firewall Settings:** Allow all incoming connections[5]

Currently, no other software has been installed. We will therefore proceed with a review of the software requirements necessary to ensure everything functions correctly. Here is the list of software requirements (based on the information provided in the clarin-dspace repository):

- **Java JDK**, version 17.0.7
- **Apache Maven**, version 3.8.1
- **Ant**, version 1.10.15
- **PostgreSQL**, version 17.5
- **Solr**, version 8.11.4
- **Tomcat**, 9
- **Git**
- **Node.js**, version 16.20.0
- **Yarn**, install after node with command npm install --global yarn
- **Apache Web Server**, 2.4
- **LetsEncrypt Certbot**
- **OpenSSL**
- **Shibboleth 3.3.0**
- **DSPACE Backend (clarin-v7)**
- **DSPACE Frontend (clarin-v7)**

If any of these software packages are already present on your machine, make sure to remove them or install other versions (if possible) and ensure that the configurations point to the desired versions.

[5] This applies only to the internal Ubuntu environment, but from a system perspective, the server is connected to the network, and access is restricted to VPN connections. Only ports 80 and 443 are open to the public, but they can only be accessed if a server is configured to listen on those ports and is set up to redirect traffic to the appropriate applications.

3. Installation

In this section, we will cover the actual installation of the DSPACE software, with all steps explained in a step-by-step manner. However, we will first proceed with the installation of the previously listed programs to ensure the proper functioning of the application.

Before starting, it's important to consider the information provided on the official DSpace website. Starting with version 7 (and above), the DSpace application is divided into two main components:

- **DSpace Frontend**[6]: the frontend consists of a User Interface built using Angular; it is a Node.js web application that requires Node.js to run after being built/compiled.
- **DSpace Backend**: the backend consists of a Server API ("server" web application) developed using Spring Boot; it is a Java web application that can operate independently but does not include a user interface.

3.1 Installing Backend Requirements

As a starting point, this guide assumes that the user connects to the target machine via SSH, allowing them to perform all operations through the command line interface. This represents the initial step from which you can proceed with the following tasks. Once the user connects via SSH, they must elevate their privileges to superuser. Therefore, the next step after establishing the encrypted connection is to enable superuser mode using the command `sudo su`.

In summary:

1. Connect to the target machine via SSH

   ```
   ssh username@machine_address.domain
   ```

2. Elevate to superuser privileges by running:

   ```
   sudo su
   ```

3.1.1 Installing Oracle Java JDK (17.0.7)

The first step is to install Oracle Java JDK version 17.0.7. To do this via the command line, we will need `curl` to download the source files[7].

[6] The frontend cannot operate independently, as it depends on a fully functional DSpace Backend. It provides all user-facing functionality and acts as the primary interface for end users.

[7] If curl is not installed on your system, please install it by following the appropriate guides for your operating system. You can find instructions for installing curl on Windows, macOS, and Linux at the following link: https://help.ubidots.com/en/articles/2165289-learn-how-to-install-run-curl-on-windows-macosx-linux

Once connected via SSH, you will find yourself in the following location: `/home/<your_username>`.

Now, we need to download the Java sources. Use the following command[8]:

```
curl -O https://download.oracle.com/java/17/archive/jdk-17.0.7_linux-x64_bin.tar.gz
```

This will download the Java JDK archive into your current directory. To extract the downloaded archive, use the following command:

```
tar -xzvf jdk-17.0.7_linux-x64_bin.tar.gz
```

Now, the next step is to move the extracted folder to the designated system directory for hosting Java versions. For Ubuntu, this directory is typically located at:

```
/usr/lib/jvm/
```

If the folder does not exist, you need to create it through the mkdir command:

```
mkdir -p /usr/lib/jvm/
```

Then, move the extracted JDK folder:

```
mv jdk-17.0.7.jdk /usr/lib/jvm/
```

After moving the JDK folder to `/usr/lib/jvm/`, the next step is to configure your system to use this Java version as the default. This is done using the `update-alternatives` command, which allows you to manage different versions of software like Java.

Run the following commands to register the newly installed Java runtime and compiler:

```
sudo update-alternatives --install /usr/bin/java java /usr/lib/jvm/jdk-17.0.7/bin/java 1
sudo update-alternatives --install /usr/bin/javac javac /usr/lib/jvm/jdk-17.0.7/bin/javac 1
```

To select the version to use, run:

```
sudo update-alternatives --config java
sudo update-alternatives --config javac
```

[8] Ensure you are in super user mode.

To make your Java installation globally available and properly configured for development tools, it's recommended to set the JAVA_HOME environment variable.

You can do this by adding the following lines to your `~/.bashrc` file (for the current user). To edit it using **vim**[9]:

```
vi ~/.bashrc
```

Once inside vim, press i to enter **insert mode**, then add the following lines at the end of the file:

```
export JAVA_HOME=/usr/lib/jvm/jdk-17.0.7
export PATH=$JAVA_HOME/bin:$PATH
```

Then press Esc, type :wq and press Enter to save and exit.

To apply the changes immediately:

```
source ~/.bashrc
```

Run the following commands to verify your configuration:

```
java -version
javac -version
echo $JAVA_HOME
```

If you have completed this step successfully, let's move on to the next step.

3.1.2 Installing Apache Maven (3.8.1)

To proceed with this installation, first download the source files using the following command:

```
curl -O https://repo.maven.apache.org/maven2/org/apache/maven/apache-maven/3.8.1/apache-maven-3.8.1-bin.tar.gz
```

After downloading, you can extract the archive with:

```
tar -xzvf apache-maven-3.8.1-bin.tar.gz
```

[9] https://www.vim.org/download.php

After extracting the archive, you should find the folder `apache-maven-3.8.1`.

Then, move the folder to the `/opt` location:

```
sudo mv apache-maven-3.8.1 /opt
```

On Ubuntu 24.04, setting up environment variables is essential to run the Maven correctly. Therefore, you are required to create a “maven.sh” file to set the environment variables:

```
sudo nano /etc/profile.d/maven.sh
```

Inside the “maven.sh” script file, copy the below lines of code:

```
export JAVA_HOME=/usr/lib/jvm/jdk-17.0.7
export M3_HOME=/opt/apache-maven-3.8.1
export MAVEN_HOME=/opt/apache-maven-3.8.1
export PATH=${M3_HOME}/bin:${PATH}
```

Run the following command to make the “maven.sh” file executable:

```
sudo chmod +x /etc/profile.d/maven.sh
```

Finally, you are required to execute the “maven.sh” file and apply changes without rebooting your system.:

```
source /etc/profile.d/maven.sh
```

To ensure the Maven package is installed correctly, check its version:

```
mvn -version
```

You should see output similar to this:

```
Apache Maven 3.8.1 (05c21c65bdfed0f71a2f2ada8b84da59348c4c5d)
Maven home: /opt/apache-maven-3.8.1
Java version: 17.0.7, vendor: Oracle Corporation, runtime: /usr/lib/jvm/jdk-17.0.7
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-62-generic", arch: "amd64", family: "unix"
```

3.1.2 Installing Apache Ant (1.10.15)

To install Apache Ant, the first step is to download the binaries directly from the Apache repository (in the Download folder). Use the following command:

```
curl -O https://dlcdn.apache.org//ant/binaries/apache-ant-1.10.15-bin.tar.gz
```

Extract the downloaded archive using the tar command:

```
tar -xzvf apache-ant-1.10.15-bin.tar.gz
```

Move the extracted folder to a system directory (e.g., /opt/ for Linux):

```
mv apache-ant-1.10.15 /opt/
```

Go to the user folder and add the Ant environment variables to your profile file trough this command:

```
vi .bashrc
```

Add the following lines:

```
export ANT_HOME=/opt/apache-ant-1.10.15
export PATH=$ANT_HOME/bin:$PATH
```

Apply the changes to the environment variables:

```
source .bashrc
```

Test the Ant installation by checking its version:

```
ant -version
```

You should see something like this:

```
Apache Ant(TM) version 1.10.15 compiled on August 25 2024
```

Now, let's proceed with the installation of PostgreSQL.

3.1.3 Installing PostgreSQL (17)

You can proceed with the PostgreSQL installation or any other necessary steps.

First, update the package index and install required packages:

```
sudo apt update
```

Add the PostgreSQL 17 repository:

```
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
```

Import the repository signing key:

```
curl -fsSL https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/postgresql.gpg
```

Update the package list:

```
sudo apt update
```

Install PostgreSQL 17 and contrib modules:

```
sudo apt install postgresql-17
```

Start and enable PostgreSQL service:

```
sudo systemctl start postgresql
sudo systemctl enable postgresql
```

Test whether PostgreSQL is now accessible by checking its version:

```
psql --version
```

You should see output similar to:

```
psql (PostgreSQL) 17.5 (Ubuntu 17.5-1.pgdg24.04+1)
```

3.1.3.1 Enable TCP/IP connections

Now, according to the DSPACE wiki, it is necessary to make changes to the configuration file to enable TCP/IP connections. To do this, you need to modify the postgresql.conf file, which on macOS is located at:

```
/etc/postgresql/17/main/postgresql.conf
```

You can edit the file using a text editor, for example:

```
sudo vi  /etc/postgresql/17/main/postgresql.conf
```

In the `postgresql.conf` file, locate and **uncomment** the line that begins with:

```
listen_addresses = 'localhost'
```

Next, enhance security by editing the `pg_hba.conf` file and adding the following line:

```
host dspace dspace 127.0.0.1 md5
```

⚠ **Ensure this line is placed before any entries that apply to all databases, as the first matching rule takes precedence.**

Explanation:

1. **`host`**:
   Specifies that this rule applies to connections over TCP/IP (as opposed to local socket connections).

2. **`dspace`** (first occurrence):
   Refers to the database name. This rule applies only to connections targeting the dspace database.

3. **`dspace`** (second occurrence):
   Refers to the username. Only the user dspace can authenticate using this rule.

4. **`127.0.0.1`**:
   Specifies that this rule applies only to connections coming from the local machine (localhost) over the loopback IP address.

5. **`md5`**:
   Defines the authentication method. md5 requires the user to authenticate using a password, which is hashed with MD5 before being sent over the network.

Now let's proceed with the installation of Apache Solr.

3.1.4 Installing Solr (8.11)

To install Solr on Ubuntu, you need to download the binaries with this command:

```
Wget https://downloads.apache.org/lucene/solr/8.11.4/solr-8.11.4.tgz
```

Extract the downloaded .tgz file:

```
tar -xvzf solr-8.11.4.tgz
```

Use the provided install script to install Solr as a systemd service:

```
./solr-8.11.4/bin/install_solr_service.sh solr-8.11.4.tgz
```

You should see a response that includes HTTP headers and a status code 200 OK, confirming that Solr is active and listening on port 8983. If you don't see this, check the Solr logs for any errors.

Now, let's proceed with the installation of Tomcat.

3.1.5 Installing Tomcat (8.11)

To improve security, create a dedicated group and system user to run the Apache Tomcat service. Create a new /opt/tomcat directory for your Tomcat installation:

```
sudo mkdir -p /opt/tomcat
```

To create a new group called **tomcat**, enter:

```
sudo groupadd tomcat
```

Create a system user named **tomcat**, assign it to the **tomcat** group, and set */opt/tomcat* as the home directory:

```
sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

Now, download the binaries:

```
curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.106/bin/apache-tomcat-9.0.106.tar.gz
```

Extract tar.gz file:

```
tar xzvf apache-tomcat-9.0.106.tar.gz -C /opt/tomcat
```

Modify Tomcat User Permission:

```
sudo chown -RH tomcat: /opt/tomcat
```

Change script permissions to ensure all scripts in the /opt/tomcat/bin/ directory are executable:

```
sudo sh -c 'chmod +x /opt/tomcat/bin/*.sh'
```

To use Tomcat as a service, create a systemd unit file. Use a text editor like Nano to create a tomcat.service file in the /etc/systemd/system directory:

```
sudo nano /etc/systemd/system/tomcat.service
```

Add these lines to tomcat.service file:

```
[Unit]
Description=Apache Tomcat Web Application Container
```

```
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

Environment="JAVA_HOME=/usr/lib/jvm/jdk-17.0.7"
Environment="CATALINA_BASE=/opt/tomcat"
Environment="CATALINA_HOME=/opt/tomcat"
Environment="CATALINA_PID=/opt/tomcat/temp/tomcat.pid"
Environment="JAVA_OPTS=-Djava.security.egd=file:///dev/urandom -Djava.awt.headless=true"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```

Reload the system daemon:

```
sudo systemctl daemon-reload
```

Start the Tomcat service:

```
sudo systemctl start tomcat
```

Enable Tomcat to start on boot:

```
sudo systemctl enable tomcat
```

Verify the Apache Tomcat service is running:

```
sudo systemctl status tomcat
```

You should see this:

```
tomcat.service - Apache Tomcat Web Application Container
     Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-06-23 13:12:53 UTC; 10min ago
   Main PID: 9019 (java)
      Tasks: 31 (limit: 11489)
     Memory: 119.7M (peak: 120.2M)
        CPU: 4.169s
     CGroup: /system.slice/tomcat.service
```

To ensure Tomcat has enough memory to run DSpace and uses UTF-8 as its default file encoding for international character support, you need to set the appropriate environment variables

Set this environment variable inside the .bashrc file and source:

```
export JAVA_OPTS="-Xmx512M -Xms64M -Dfile.encoding=UTF-8"
source .bashrc
```

Modify server.xml to set URIEncoding in UTF-8:

```
vi /opt/tomcat/conf/server.xml
```

Then change the line where start the Connector node:

```
<Connector port="8080"
               minSpareThreads="25"
               enableLookups="false"
               redirectPort="8443"
               connectionTimeout="20000"
               disableUploadTimeout="true"
               URIEncoding="UTF-8"/>
```

Restart Tomcat service using:

```
systemctl restart tomcat
```

This concludes the final step for installing the minimum requirements for the backend. Now, let's proceed with the necessary steps to install the backend.

3.1.5.1 Change log verbosity

To reduce the verbosity of Apache Tomcat logs and limit informational messages, the logging level was changed from `INFO` to `WARN` for both file-based and console handlers.

The procedure was performed as follows:

1. Administrative privileges were obtained by switching to the superuser:

   ```
   sudo su
   ```

2. The Tomcat configuration directory was accessed:

   ```
   cd /opt/tomcat/conf
   ```

3. The Tomcat logging configuration file was opened for editing:

```
vi logging.properties
```

4. Within the logging.properties file, the logging level was modified from INFO to WARN for the following asynchronous file handlers:

   - **Catalina handler**

```
1catalina.org.apache.juli.AsyncFileHandler.level = WARN
1catalina.org.apache.juli.AsyncFileHandler.directory = ${catalina.base}/logs
1catalina.org.apache.juli.AsyncFileHandler.prefix = catalina.
1catalina.org.apache.juli.AsyncFileHandler.maxDays = 90
1catalina.org.apache.juli.AsyncFileHandler.encoding = UTF-8
```

   - **Localhost handler**

```
2localhost.org.apache.juli.AsyncFileHandler.level = WARN
2localhost.org.apache.juli.AsyncFileHandler.directory = ${catalina.base}/logs
2localhost.org.apache.juli.AsyncFileHandler.prefix = localhost.
2localhost.org.apache.juli.AsyncFileHandler.maxDays = 90
2localhost.org.apache.juli.AsyncFileHandler.encoding = UTF-8
```

   - **Manager handler**

```
3manager.org.apache.juli.AsyncFileHandler.level = WARN
3manager.org.apache.juli.AsyncFileHandler.directory = ${catalina.base}/logs
3manager.org.apache.juli.AsyncFileHandler.prefix = manager.
3manager.org.apache.juli.AsyncFileHandler.maxDays = 90
3manager.org.apache.juli.AsyncFileHandler.encoding = UTF-8
```

   - **Host-manager handler**

```
4host-manager.org.apache.juli.AsyncFileHandler.level = WARN
4host-manager.org.apache.juli.AsyncFileHandler.directory = ${catalina.base}/logs
4host-manager.org.apache.juli.AsyncFileHandler.prefix = host-manager.
4host-manager.org.apache.juli.AsyncFileHandler.maxDays = 90
4host-manager.org.apache.juli.AsyncFileHandler.encoding = UTF-8
```

5. The logging level for the console handler was also updated:

```
java.util.logging.ConsoleHandler.level = WARN
java.util.logging.ConsoleHandler.formatter = org.apache.juli.OneLineFormatter
java.util.logging.ConsoleHandler.encoding = UTF-8
```

6. After applying the changes, the file was saved and the editor was closed.

7. The Tomcat service was restarted to apply the new logging configuration:

```
systemctl restart tomcat
```

8. Finally, the service status was checked to ensure that Tomcat started correctly:

```
systemctl status tomcat
```

This configuration reduces log noise by suppressing informational messages, while still preserving warnings and error-level events relevant for system monitoring and troubleshooting.

3.1.6 Installing Node JS (16)

Run the following command to install the latest version of NVM (v0.40.3 at the time of writing):

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.3/install.sh | bash
```

After installation, you need to load NVM into your current shell session:

```
\. "$HOME/.nvm/nvm.sh"
```

Use NVM to install **Node.js version 16**, which will also install the corresponding version of npm.

```
nvm install 16
```

Check version:

```
node -v
```

You should see:

```
v16.20.2
```

3.1.7 Installing Git

To install Git, simply run:

```
sudo apt-get install git-all
```

3.1.8 Installing Yarn

Run this command:

```
npm install --global yarn
```

3.1.9 Installing LetsEncrypt

This section describes the procedure to install and configure **Let's Encrypt** SSL certificates on a Linux server (Ubuntu-based), using the **Certbot** tool. Let's Encrypt provides free, automated, and trusted certificates for HTTPS support. Certbot is the recommended client to interact with the Let's Encrypt Certificate Authority (CA).

Update the package list and install **Certbot** and the appropriate plugin (e.g. Apache or Nginx):

```
sudo apt install certbot python3-certbot-apache -y
```

3.1.10 Installing Shibboleth

The Shibboleth Service Provider is a daemon designed to handle federated authentication using the **SAML2 protocol**. It is widely adopted within the CLARIN infrastructure for implementing federated Single Sign-On (SSO). The following steps describe how to compile and install Shibboleth SP from source along with all required dependencies.

Install all system packages required for building the software stack:

```
sudo apt update
sudo apt install -y \
  gcc g++ make libssl-dev libcurl4-openssl-dev autoconf automake \
  libtool python3-magic python3-dateutil python3-lxml pkg-config apache2-dev
```

We define the following standard paths:

- **Installation directory**: /opt/shibboleth-sp ($SHIB_DIR)
- **Source/dependencies directory**: /opt/shibboleth-sw ($SHIB_SW)

Create the directories if they do not exist:

```
sudo mkdir -p /opt/shibboleth-sp
sudo mkdir -p /opt/shibboleth-sw
```

3.1.10.1 Building Shibboleth Dependencies

All third-party dependencies will be downloaded and built inside $SHIB_SW.

- **Boost (1.79.0)**

```
cd /tmp
wget https://boostorg.jfrog.io/artifactory/main/release/1.79.0/source/boost_1_79_0_rc1.tar.gz -O
boost.tar.gz
cd /opt/shibboleth-sw
tar -xvzf /tmp/boost.tar.gz
```

- **log4shib (2.0.1)**

```
cd /tmp
wget https://shibboleth.net/downloads/log4shib/2.0.0/log4shib-2.0.1.tar.gz -O log4shib.tar.gz
cd /opt/shibboleth-sw
tar -xvzf /tmp/log4shib.tar.gz
cd log4shib-2.0.1
./configure --disable-static --disable-doxygen --prefix=/opt/shibboleth-sp
sudo make
sudo make install
```

- **zlib (1.31)**

```
cd /tmp
wget http://www.zlib.net/zlib-1.2.11.tar.gz -O zlib.tar.gz
cd /opt/shibboleth-sw
tar -xvzf /tmp/zlib.tar.gz
cd zlib-1.2.11
./configure --prefix=/opt/shibboleth-sp
sudo make
sudo make install
```

- **Xerces-C (3.3.0)**

```
cd /tmp
wget https://mirror.nohup.it/apache//xerces/c/3/sources/xerces-c-3.2.3.tar.gz -O xerces.tar.gz
cd /opt/shibboleth-sw
tar -xvzf /tmp/xerces.tar.gz
cd xerces-c-3.2.3
./configure --prefix=/opt/shibboleth-sp --disable-netaccessor-libcurl
sudo make
sudo make install
```

- **xml-security-c (2.0.4)**

```
cd /tmp
wget https://dlcdn.apache.org/santuario/c-library/xml-security-c-2.0.4.tar.gz -O xmlsec.tar.gz
cd /opt/shibboleth-sw
tar -xvf xmlsec.tar.gz
cd /opt/shibboleth-sw/xml-security-c-2.0.4
```

```
sudo ./configure --without-xalan --disable-static --prefix=/opt/shibboleth-sp --with-
openssl=/usr/include/openssl
sudo make
sudo make install
```

If errors occur:

```
sudo ./configure --without-xalan --disable-static --prefix=/opt/shibboleth-sp --with-
openssl=/usr/include/openssl xerces_CFLAGS='-I/opt/shibboleth-sp/include' XERCES_LDFLAGS=''
xerces_LIBS='-L/opt/shibboleth-sp/lib -lxerces-c'
sudo make
sudo make install
```

- **xmltooling (3.2.1)**

```
cd /tmp
wget http://shibboleth.net/downloads/c++-opensaml/latest/xmltooling-3.2.1.tar.gz -O xmltool.tar.gz
cd /opt/shibboleth-sw
tar -xvf xmltool.tar.gz
cd /opt/shibboleth-sw/xmltooling-3.2.1
sudo ./configure --prefix=/opt/shibboleth-sp -C --with-boost=/opt/shibboleth-sw/boost_1_79_0_rc1
make
make install
```

If errors occur:

```
sudo ./configure --prefix=/opt/shibboleth-sp -C --with-boost=/opt/shibboleth-sw/boost_1_79_0_rc1
log4shib_CFLAGS='-I/opt/shibboleth-sp/include' log4shib_LIBS='-L/opt/shibboleth-sp/lib -llog4shib'
xerces_CFLAGS='-I/opt/shibboleth-sp/include' xerces_LDFLAGS='' xerces_LIBS='-L/opt/shibboleth-sp/lib -
lxerces-c' xml_security_CFLAGS='-I/opt/shibboleth-sp/include' xml_security_LIBS='-L/opt/shibboleth-
sp/lib -lxml-security-c'
sudo make
sudo make install
```

- **OpenSAML (3.2.1)**

```
cd /tmp
wget http://shibboleth.net/downloads/c++-opensaml/latest/opensaml-3.2.1.tar.gz -O opensaml.tar.gz
cd /opt/shibboleth-sw
tar -xvf opensaml.tar.gz
cd /opt/shibboleth-sw/opensaml-3.2.1
sudo ./configure --with-log4shib=/opt/shibboleth-sp --prefix=/opt/shibboleth-sp -C --with-
boost=/opt/shibboleth-sw/boost_1_79_0_rc1
sudo make
sudo make install
```

If errors occur:

```
sudo ./configure --prefix=/opt/shibboleth-sp -C --with-boost=/opt/shibboleth-sw/boost_1_79_0_rc1
log4shib_CFLAGS='-I/opt/shibboleth-sp/include' log4shib_LIBS='-L/opt/shibboleth-sp/lib -llog4shib'
xerces_CFLAGS='-I/opt/shibboleth-sp/include' xerces_LDFLAGS='' xerces_LIBS='-L/opt/shibboleth-sp/lib -
lxerces-c' xmlsec_CFLAGS='-I/opt/shibboleth-sp/include' xmlsec_LIBS='-L/opt/shibboleth-sp/lib -lxml-
security-c' xmltooling_CFLAGS='-I/opt/shibboleth-sp/include' xmltooling_LIBS='-L/opt/shibboleth-sp/lib
-lxmltooling'
sudo make
sudo make install
```

- **Shibboleth-SP (3.3.0)**

```
cd /tmp
wget https://shibboleth.net/downloads/service-provider/latest/shibboleth-sp-3.3.0.tar.gz -O shib.tar.gz
cd /opt/shibboleth-sw
tar -xvf shib.tar.gz
```

Create symbolic link with boost:

```
ln -s /opt/shibboleth-sw/boost_1_79_0/boost /opt/shibboleth-sw/shibboleth-sp-3.3.0/shibsp/boost
```

Build and install:

```
cd /opt/shibboleth-sw/shibboleth-sp-3.3.0
sudo ./configure --with-log4shibv2=/opt/shibboleth-sp --prefix=/opt/shibboleth-sp -C --with-
boost=/opt/shibboleth-sw/boost_1_79_0_rc1
sudo make
sudo make install
```

If errors occur:

```
./configure --prefix=/opt/shibboleth-sp --with-apxs=/usr/bin/apxs --enable-apache-24 --with-
apxs24=/usr/bin/apxs --with-boost=/opt/shibboleth-sw/boost_1_79_0_rc1 log4shib_CFLAGS='-
I/opt/shibboleth-sp/include' log4shib_LIBS='-L/opt/shibboleth-sp/lib -llog4shib' xerces_CFLAGS='-
I/opt/shibboleth-sp/include' xerces_LIBS='-L/opt/shibboleth-sp/lib -lxerces-c' xmlsec_CFLAGS='-
I/opt/shibboleth-sp/include' xmlsec_LIBS='-L/opt/shibboleth-sp/lib -lxml-security-c'
xmltooling_CFLAGS='-I/opt/shibboleth-sp/include' xmltooling_LIBS='-L/opt/shibboleth-sp/lib -
lxmltooling' xmltooling_lite_CFLAGS='-I/opt/shibboleth-sp/include' xmltooling_lite_LIBS='-
L/opt/shibboleth-sp/lib -lxmltooling-lite' opensaml_CFLAGS='-I/opt/shibboleth-sp/include'
opensaml_LIBS='-L/opt/shibboleth-sp/lib -lsaml'
sudo make
sudo make install
```

3.1.10.2 Start Shibboleth

The Shibboleth daemon (shibd) is located in /opt/shibboleth-sp/sbin/ .

3.1.10.3 Shibboleth as a service

Create the `systemd` service definition for the Shibboleth Service Provider daemon:

```
sudo nano /etc/systemd/system/shibd.service
```

Insert the following configuration:

```
[Unit]
Description=Shibboleth Service Provider Daemon
After=network.target

[Service]
Type=forking

ExecStart=/opt/shibboleth-sp/sbin/shibd -f \
  -c /opt/shibboleth-sp/etc/shibboleth/shibboleth2.xml \
  -p /opt/shibboleth-sp/shib_pid/shibd.pid \
  -w 30 \
  -x /opt/shibboleth-sp/share/xml/xmltooling/catalog.xml:\
/opt/shibboleth-sp/share/xml/shibboleth/catalog.xml:\
/opt/shibboleth-sp/share/xml/opensaml/saml20-catalog.xml:\
/opt/shibboleth-sp/share/xml/opensaml/saml11-catalog.xml

PIDFile=/opt/shibboleth-sp/shib_pid/shibd.pid

ExecReload=/bin/kill -HUP $MAINPID
ExecStop=/bin/kill -TERM $MAINPID

User=root
Group=root
WorkingDirectory=/opt/shibboleth-sp

Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Save the file and exit the editor.

Reload the `systemd` daemon to apply the new service definition:

```
sudo systemctl daemon-reload
```

Start the Shibboleth Service Provider daemon:

```
sudo systemctl start shibd
```

Check the status of the service to ensure it is running correctly:

```
sudo systemctl status shibd -l
```

The output should indicate that the service is **active (running)**.

Enable the Shibboleth service to start automatically at system boot:

```
sudo systemctl enable shibd
```

This configuration ensures that the Shibboleth Service Provider daemon is properly managed by systemd, automatically started at boot time, and restarted in case of failures—providing a stable foundation for federated authentication in a DSpace deployment.

3.2 Installing DSPACE (backend + frontend)

After successfully installing all the main components required for the DSPACE installation, we can now proceed with installing the core software. However, before doing so, we need to complete some preliminary steps to further prepare an environment suitable for running the software.

You have to create a folder named shibd_pid where store the pid file for the process.

```
cd /opt/shibboleth-sp/
mkdir shibd_pid
```

To run Shibboleth, go to sbin folder of the shibboleth software and run this command:

```
./shibd -f -c /opt/shibboleth-sp/etc/shibboleth/shibboleth2.xml -p /opt/shibboleth-
sp/shib_pid/shibd.pid -w 30 -x /opt/shibboleth-sp/share/xml/xmltooling/catalog.xml:/opt/shibboleth-
sp/share/xml/shibboleth/catalog.xml:/opt/shibboleth-sp/share/xml/opensaml/saml20-
catalog.xml:/opt/shibboleth-sp/share/xml/opensaml/saml11-catalog.xml
```

To check if the Shibboleth daemon (shibd) is running, you can use the following command:

```
ps aux | grep "shib[d]"
```

3.2.1 Create a DSPACE system user

The first step is to create a dedicated operating system user for DSpace. This user will have full read/write access to the DSpace installation directory, which is a recommended security measure.

To create a new user account for DSpace, use the following command:

```
useradd -m dspace
```

3.2.2 Downloading the DSPACE Sources

Now, you need to download the DSpace source files and place them under the user directory. This path will henceforth be referred to as `[dspace-source-folder]`.

Download the source code using this git command:

```
git clone https://github.com/ufal/clarin-dspace.git
```

Enter the folder using the command:

```
cd clarin-dspace
```

By default, when downloading the DSpace project, the main branch is named `clarin`. However, we need the `clarin-v7` branch. To switch to this branch, use the following command:

```
git checkout clarin-v7
```

After completing these steps, we will now focus on the next task: configuring the database.

3.2.3 Database Setup

The first step is create the `dspace` database user using this command:

```
sudo -u postgres createuser --username=postgres --no-superuser --pwprompt dspace
```

You will be asked to enter a password for the new dspace user twice to confirm it. Please, enter “dspace” value.

Set up a `dspace` database and assign ownership to the `dspace` PostgreSQL user. As with the previous step, this action requires a PostgreSQL "superuser" account.

```
sudo -u postgres createdb --username=postgres --owner=dspace --encoding=UNICODE dspace
```

Lastly, it is essential to activate the pgcrypto extension in your newly created dspace database. This action, like before, requires a PostgreSQL "superuser" account.

```
sudo -u postgres psql -d dspace -c "CREATE EXTENSION pgcrypto;"
```

3.2.4 Create DSPACE folder

Since initial dpsace.cfg file set the dspace folder to the root level, let’s make the folder using super user privileges:

```
cd /
```

Create a folder `dspace`:

```
sudo mkdir dspace
```

3.2.5 Building the DSPACE Sources

To do this, place yourself in the root of `clarin-dspace` folder and launch:

```
mvn clean install
```

Go down to `/dspace/target/dspace-installer`:

```
cd /dspace/target/dspace-installer
```

Use the command:

```
ant fresh_install
```

You should see a screen like:

```
init_spiders:
     [echo]
     [echo] ====================================================================
     [echo]  The DSpace code has been installed.
     [echo]
     [echo]  To complete installation, you should do the following:
     [echo]
     [echo]  * Setup your Web servlet container (e.g. Tomcat) to look for your
     [echo]    DSpace web applications in: /dspace/webapps/
     [echo]
     [echo]    OR, copy any web applications from /dspace/webapps/ to
     [echo]    the appropriate place for your servlet container.
     [echo]    (e.g. '$CATALINA_HOME/webapps' for Tomcat)
     [echo]
     [echo]  * Start up your servlet container (e.g. Tomcat). DSpace now will
     [echo]    initialize the database on the first startup.
     [echo]
     [echo]  * Make an initial administrator account (an e-person) in DSpace:
     [echo]
     [echo]    /Users/gruppoh2ioscicloud/dspace/bin/dspace create-administrator
     [echo]
     [echo]  You should then be able to access your DSpace's REST API:
     [echo]
     [echo]    http://localhost:8080/server
     [echo]
     [echo] ====================================================================
     [echo]
```

It means the build was ended correctly.

After, give all permission to the dspace folder to tomcat user, in order to deploy the web app:

```
sudo chown -R tomcat:tomcat /dspace
```

3.2.6 Copy Solr Cores

Copy the Solr cores from the dspace build folder to the Solr user directory placed at /var/solr/data.

First, you have to create the configsets folder. Then, run as sudo user:

```
sudo mkdir /var/solr/data/configsets
```

After, you have to copy the Solr cores with:

```
sudo cp -R /dspace/solr/* /var/solr/data/configsets
```

Then, assign all permissions to solr user:

```
sudo chown -R solr:solr /var/solr/data/configsets/*
```

Restart Solr:

```
sudo systemctl restart solr
```

3.2.7 Check database connection

In order to check the database connection navigate to the DSpace installation directory located at the following path and run this command:

```
cd /dspace/bin
./dspace database migrate
```

You should see this screen:

```
Database URL: jdbc:postgresql://localhost:5432/dspace
Migrating database to latest version... (Check dspace logs for details)
Done.
```

3.2.8 Create admin

To create the system administrator, navigate to the DSpace installation directory located at the following path:

```
/dspace/bin
```

Launch this command:

```
./dspace create-administrator
```

The system will prompt you to provide the following information:

- Email address
- First name
- Select one of the following languages (e.g., en for English)
- Confirmation of the entered data
- Password (I used "admin")

```
dev-clarin@dev-clarin dspace % bin/dspace create-administrator
Creating an initial administrator account
E-mail address: michele.mallia@cnr.it
First name: Michele
Last name: Mallia
Select one of the following languages: en
Language: en
Is the above data correct? (y or n): y
Password will not display on screen.
Password:
Again to confirm:
Administrator account created
```

3.2.9 Deploy the Server webapp

You have to copy the content of the DSPACE builded code to the tomcat webapp folder.

So, launch this command:

```
sudo cp -R /dspace/webapps/* /opt/tomcat/webapps/
```

Ensure the entire folder is owned by tomcat user:

```
sudo chown -R tomcat:tomcat /dspace
```

Ensure to reload the tomcat service using systemctl command.

3.2.10 Deploy the Frontend sources

In order to deploy the frontend sources, you have to download the source code. Please go to the user folder and clone the repository:

```
git clone git@github.com:ufal/dspace-angular.git
```

Go to the dspace-angular folder and run this command:

```
yarn install
```

After, run this command:

```
yarn build:prod
```

After the build is complete, launch the frontend web app through this command:

```
node ./dist/server/main.js
```

You can view the frontend GUI going to localhost:4000.

4. Configuration

After completing the installation of all required system components and software dependencies, the next phase involves configuring the environment to support a functional and stable DSpace 7 installation.

4.1 Generate LetsEncrypt Certificate

To ensure secure HTTPS connections, a TLS/SSL certificate must be obtained and properly installed. Let's Encrypt provides free, automated, and trusted certificates suitable for production environments. The recommended tool for obtaining and renewing certificates is **Certbot**.

Prerequisites

- A public domain name (e.g., dev-clarin.ilc.cnr.it) correctly configured to point to the server IP.
- TCP ports **80** (HTTP) and **443** (HTTPS) open and accessible from the internet.
- A working web server (e.g., Apache or Nginx) already installed and running.

On Ubuntu 24.04 LTS, install Certbot and the Apache plugin with:

```
sudo apt update
sudo apt install certbot python3-certbot-apache
```

Ensure that Apache service is down, then run the following command to automatically request and configure the SSL certificate for your domain:

```
sudo certbot certonly --apache -d dev-clarin.ilc.cnr.it
```

During the process, Certbot will:

- Communicate with the Let's Encrypt CA to validate domain ownership.
- Install the certificate into the Apache configuration.
- Optionally enable automatic HTTP-to-HTTPS redirection.

If successful, Certbot will confirm the location of your certificate and private key:

```
/etc/letsencrypt/live/dev-clarin.ilc.cnr.it/fullchain.pem
/etc/letsencrypt/live/dev-clarin.ilc.cnr.it/privkey.pem
```

4.2 Configure LetsEncrypt Certificate on Apache

Once the Let's Encrypt certificate has been successfully obtained using Certbot, Apache must be properly configured to use the certificate for HTTPS connections.

Edit or create the Apache configuration file for your domain (e.g. /etc/apache2/sites-available/yourdomain.conf), and ensure that the <VirtualHost *:443> section contains the correct certificate paths and HTTPS settings:

Create a configuration folder “dspace-conf” and copy all the files from sites-available (you should find two files, default-ssl.conf and 000-default.conf).

Modify the default-ssl.conf with this configuration:

```
<IfModule mod_shib>
    <VirtualHost *:443>
        ServerAdmin michele.mallia@cnr.it
        ServerName dev-clarin.ilc.cnr.it
        DocumentRoot /var/www/dev-clarin


        # CREATE A FOLDER IN THE PATH
        ScriptAlias /cgi-bin/ /var/www/dev-clarin/secure/

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined


        SSLEngine on

        SSLCertificateFile      /etc/letsencrypt/live/dev-clarin.ilc.cnr.it/fullchain.pem
        SSLCertificateKeyFile   /etc/letsencrypt/live/dev-clarin.ilc.cnr.it/privkey.pem


        # handle cmdi requests with redirects
        RewriteCond %{HTTP_ACCEPT} (.*cmdi.xml.*)
        RewriteRule ^(.*)/handle/(.*)$ https://%{HTTP_HOST}/server/cmdi/oai-
metadata?metadataPrefix=cmdi&handle=$2 [L,R=301]
        RewriteCond %{QUERY_STRING} format=cmdi [NC]
        RewriteRule ^(.*)/handle/(.*)$ https://%{HTTP_HOST}/server/cmdi/oai-
metadata?metadataPrefix=cmdi&handle=$2 [L,R=301]


        RewriteCond %{REQUEST_URI} ^/repository$
        RewriteRule (.*) /repository/xmlui [R,L]



        XSendFile On
        XSendFilePath /opt/dspace/assetstore/

        LogLevel info ssl:debug

        ErrorLog ${APACHE_LOG_DIR}/dev-clarin-error.log
        CustomLog ${APACHE_LOG_DIR}/dev-clarin-access.log combined


        # ENSURE YOU HAVE THIS FILE
        include conf-enabled/shib-dev-clarin.conf


        ProxyPreserveHost On
        RequestHeader set X-Forwarded-Proto https
```

```
        ProxyPass /server http://localhost:8080/server
        ProxyPassReverse /server http://localhost:8080/server


        ProxyPass /Shibboleth.sso !
        ProxyPass / http://localhost:4000/
        ProxyPassReverse / http://localhost:4000/

    </VirtualHost>

</IfModule>
```

To enable Shibboleth-based authentication for your service, ensure that Apache is properly configured with the following directives.

Make sure a folder exists at the desired location to serve as the handler path for Shibboleth. For instance:

```
ScriptAlias /cgi-bin/ /var/www/dev-clarin/secure/
```

Verify that the Apache configuration includes the Shibboleth-specific virtual host configuration. This is typically placed under conf-enabled/ and may be site-specific:

```
Include conf-enabled/shib-dev-clarin.conf
```

This is the content of shib-dev-clarin.conf:

```
<Location /secure>
                AuthType shibboleth
                #ShibRequestSetting applicationId dev-clarin-override
                ShibRequireSession On
                require valid-user
</Location>

# lindat login


<Location "/server/api/authn/shibboleth">

                Order deny,allow
                Allow from all
                AuthType shibboleth
                ShibRequireSession On
                ShibUseHeaders On
                require valid-user
                require valid-user
                Require shibboleth
        </Location>



        <Directory "/piwik">
                AllowOverride Limit
                Order allow,deny
```

```
                Allow from 146.48.92.0/24
                Allow from 146.48.93.0/24
                Allow from 2a00:1620:c0:5c::/56
                DirectoryIndex index.php
        </Directory>


        <Files ~ "\.(pl|cgi)$">
                SetHandler perl-script
                PerlResponseHandler ModPerl::PerlRun
                Options +ExecCGI
                PerlSendHeader On
        </Files>


        <Location /perl2>
                  AuthType shibboleth
                  ShibRequestSetting requireSession 1
                  require shibboleth
                  ShibRequestSetting applicationId dev-clarin.ilc.cnr.it-override
                  AuthType shibboleth
                  ShibRequireSession On

                  ShibUseHeaders On
                  require valid-user
                  ShibRequestSetting applicationId dev-clarin.ilc.cnr.it-override

                  Options +ExecCGI
                  AddHandler cgi-script .php
                  DirectoryIndex shib_test.php


        </Location>

        <Location /perl>
                AuthType shibboleth
                ShibRequestSetting applicationId dev-clarin.ilc.cnr.it-override
                ShibRequireSession On
                require valid-user
        </Location>
```

4.3 Create new configuration for Apache

In this section, we detail how to create and enable a new Apache configuration file for your DSpace or Shibboleth-enabled service.

```
# View a list of available modules
a2enmod

# Enable required modules (example)
sudo a2enmod rewrite
sudo a2enmod ssl
sudo a2enmod headers
sudo a2enmod shib2
```

To maintain a clean and modular configuration structure, it is recommended to create a dedicated configuration folder:

```
cd /etc/apache2/
mkdir dspace-conf
cp sites-available/* dspace-conf
```

If you are maintaining a custom Apache structure, make sure the main Apache configuration file (/etc/apache2/apache2.conf) includes the path to your new config file, although this is usually handled automatically by a2enconf:

```
IncludeOptional dspace-conf/*.conf
```

Restart apache with:

```
sudo systemctl restart apache2
```

4.4 Generate OpenSSL Certificate for Shibboleth

In order for the Shibboleth Service Provider to securely communicate with Identity Providers (IdPs), a public/private key pair and self-signed certificate must be generated. These will be used to sign and decrypt SAML assertions.

```
sudo mkdir -p /etc/apache2/self_signed
cd /etc/apache2/self_signed
```

Use OpenSSL to generate a 2048-bit RSA private key and a self-signed certificate:

```
sudo openssl req -new -x509 -days 365 -nodes -sha256 \
  -out dev-clarin.crt -keyout dev-clarin.key
```

You will be prompted to enter some information:

- **Country Name** (e.g., IT)
- **State or Province Name** (e.g., Pisa)
- **Locality Name** (e.g., Pisa)
- **Organization Name** (e.g., ILC CNR)

- **Organizational Unit Name** (e.g., CLARIN)
- **Common Name**: must match your service hostname (e.g., dev-clarin.ilc.cnr.it)
- **Email Address**: a valid contact email

4.5 Configure etc/hosts file

In order to ensure proper name resolution during development or internal testing, you may need to manually map the server's IP address to its hostname by editing the /etc/hosts file.

Use a text editor with root privileges to edit the file:

```
sudo vi /etc/hosts
```

At the end of the file, add the following line:

```
192.168.xxx.xxx dev-clarin dev-clarin.ilc.cnr.it
```

Exit and save.

4.6 DSPACE Backend configuration

Once the DSpace backend has been installed, a series of configuration steps are required to ensure the system operates correctly in the intended environment. These configurations apply to the general setup of the backend and include database connectivity, file storage paths, authentication settings, and server parameters.

This section outlines the general configuration procedures that apply regardless of deployment specifics (development, staging, or production). The main configuration file for the DSpace backend is located at:

```
/dspace/config/dspace.cfg
```

You have to modify these important variables:

```
dspace.dir = /dspace
csvexport.dir = ${dspace.dir}/exports
dspace.server.url = https://dev-clarin.ilc.cnr.it/server
dspace.ui.url = https://dev-clarin.ilc.cnr.it
dspace.name = <your institution name>
```

```
dspace.shortname = DSpace
handle.additional.prefixes = 20.500.11752, 000-c0-111
proxies.trusted.ipranges = 192.168.xxx.xxx
```

Under the /modules folder, you will find assetstore.cfg, please modify this variable:

```
assetstore.dir = /opt/dspace/assetstore
```

Restart tomcat to load the system with the new variables.

4.6.1 Change log level of DSPACE

To reduce the verbosity of DSpace logging and limit the amount of informational output, the log level was adjusted by modifying the Log4j2 configuration file.

The procedure was performed as follows:

1. Navigate to the DSpace configuration directory:

```
cd /dspace/config
```

2. Open the Log4j2 configuration file for editing:

```
vi log4j2.xml
```

3. Locate the <Properties> section and update the logging verbosity by setting the log level to WARN for both DSpace-specific components and third-party libraries, as shown below:

```
<Properties>
    <!-- Default log file directory for DSpace.
         NOTE: The value of 'dspace.dir' will be replaced by its value in
         your configuration when DSpace is installed. -->
    <Property name="log.dir">${log4j:configParentLocation}/../log</Property>

    <!-- Log level for all DSpace-specific code (org.dspace.*)
         Possible values (from most to least verbose):
         DEBUG, INFO, WARN, ERROR, FATAL -->

    <Property name="loglevel.dspace">WARN</Property>

    <!-- Log level for other third-party tools/APIs used by DSpace
         Possible values (from most to least verbose):
         DEBUG, INFO, WARN, ERROR, FATAL -->
    <Property name="loglevel.other">WARN</Property>
</Properties>
```

In this configuration, the previous log level (INFO) was replaced with WARN in order to suppress non-critical informational messages.

4. Save the file and exit the editor.

5. Restart the Apache Tomcat service to apply the new logging configuration:

```
systemctl restart tomcat
```

After the restart, DSpace will log only warnings, errors, and fatal messages, reducing log noise while preserving information relevant for monitoring and troubleshooting.

4.7 DSPACE Frontend configuration

The DSpace frontend, built with Angular, requires a dedicated configuration phase to adapt the interface to institutional requirements and to enable authentication mechanisms such as Shibboleth (AAI). This section outlines the key steps for configuring and running the frontend environment in production.

Environment-specific configurations such as API endpoints, AAI integration, and feature toggles should be defined in the appropriate environment files (e.g., dspace-angular/config/config.production.yml). Ensure the following variables are properly set:

```
debug: false

ui:
  ssl: false
  host: localhost
  port: 4000
  nameSpace: /
  useProxies: true

rest:
  ssl: true
  host: dev-clarin.ilc.cnr.it
  port: 443
  nameSpace: /server
```

Go to dspace-angular/src/environments/environment.production.ts and put this:

```
export const environment: Partial<BuildConfig> = {
  production: true,

  ui: {
        ssl: false,
        host: 'localhost',
        port: 4000,
        nameSpace: '/',
        useProxies: true,
  },
  rest: {
        ssl: true,
        host: 'dev-clarin.ilc.cnr.it',
        port: 443,
        nameSpace: '/server',
  },
  // Angular Universal settings
  universal: {
```

```
    preboot: true,
    async: true,
    time: false,
    inlineCriticalCss: false,
  },
};
```

Customize the AAI configuration by editing the following files:

src/aai/aai.js (edit only these lines):

```
var host = 'https://' + window.location.hostname,
        ourEntityID = host.match("dev-clarin.ilc.cnr.it") ? "https://dev-clarin.ilc.cnr.it" : host;
//var namespace = 'repository';
var namespace = '';
```

src/aai/aai-config.js (edit only these lines):

```
instance.responseUrl =
                (window.location.hostname.search("dev-clarin") >= 0) ?
                        "" :
                        instance.host + instance.port + instance.repoPath +
                            "themes/UFAL/lib/html/disco-juice.html?";

...

instance.serviceName = "DEV CLARIN TEST Repository";

...

instance.localauth =
                '<form method="post" action="' + instance.target + 'login"> ' +
                    '<p>Sign in using your local account obtained from the LINDAT/CLARIAH-CZ
administrator.</p>' +
                    '<p style="margin: 5px; color: #888" ><input type="text" name="login_email"
style="font-size: 160%; width: 100%" id="login" /> <label for="login">E-Mail Address</label></p>' +
                    '<p style="margin: 5px; color: #888" ><input type="password" name="login_password"
style="font-size: 160%; width: 100%" id="pass" /> <label for="pass">Password</label></p>' +
                    '<p style="margin: 5px; color: #607890; text-decoration: underline;"><a href="' +
instance.target + 'forgot">Forgot your password?</a></p>' +
                    '<p  style="" ><input type="submit" style="margin: 20px 2px" name="submit"
value="Sign in" /></p>' +
                    '</form>';
```

To ensure that the frontend runs persistently in the background, use **PM2**, a production-grade process manager for Node.js applications:

```
npm install -g pm2
pm2 start dist/dspace-angular/server/main.js --name dspace-frontend
```

This setup ensures the frontend automatically restarts on server reboot and is monitored as a background service.

Once these steps are completed, the frontend will be fully integrated with the backend and authentication system, and ready for production deployment.

Always rebuild the frontend (ng build --configuration=production) after making changes to environment files or configuration settings.

4.7.1 Customization of graphic elements

Here I describe the procedures for customising the header and footer of DSPACE 7 to make them look like those of DSPACE 5.

First of all:

- Copied the dspace-angular folder under /opt/ilc4clarin.
- Created the file dspace-angular-cleaner.sh to clean cache and temporary files.
- Created a new theme called ilc-clarin under src/themes/ilc-clarin.
- Created the CSS/SCSS files inside the same folder.

To load the theme, it must be added to the themes section in config/config.production.yml:

```
themes:
 - name: 'ilc-clarin'
   regex: '.*'
```

Also add the following piece to the style section of angular.js:

```
{ "input": "src/themes/ilc-clarin/styles/theme.scss",
            "inject": false,
            "bundleName": "ilc-clarin-theme"
            }
```

The header to be changed is located under **src/themes/ilc-clarin/app/header/** and is called **header.component.html**. My advice is to make minimal changes to the structure, which, as it has been made available, has the Angular logic that works:

For example:

```
<a href="{{ '/' + translateSlug('partner') }}" class="lindat-dropdown-item">{{'navbar.about.partners' | translate}}</a>
```

It does two things:

1. It manages the partner page as an Angular route (i.e. it manages the automatic refresh of the page).
2. It translates the content of the navbar.about.partners variable.

PLEASE NOTE

Many static files in dspace5 are located under **/opt/git/ilc4clarin-dspace/dspace/modules/xmlui/src/main/webapp/themes/UFAL/lib/html/**.

These must be copied as HTML fragments (no HTML tags, body or head) under the src/assets/files/ folder as .html files.

Every file that is accessed by dspace5 and copied must be made an Angular component so that it is served by the application and maintains the correct style sheets.

4.7.1.1 Creation of About page (and the other components)

I will describe how to transfer the content of https://dspace-clarin-it.ilc.cnr.it/repository/xmlui/page/about to DSPACE7 and how to make it accessible from the corresponding menu item.

Proceed with creating the folder:

```
mkdir -p src/themes/ilc-clarin/app/about

touch src/themes/ilc-clarin/app/about/about.component.ts
```

```
touch src/themes/ilc-clarin/app/about/about.component.html
```

Then open the visual code editor and edit the typescript file:

about.component.ts

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-about',
  templateUrl: './about.component.html'
})
export class AboutComponent implements OnInit {
  htmlContent = '';

  constructor() {}

  ngOnInit() {
     }
}
```

You need to copy the corresponding static HTML file and insert it into the HTML file of the newly created component.

Take the files from `/src/static-files/about.html` and copy the content of the HTML file inside a `<div>` element with the classes `container` and `mt-4`, like this:

about.component.html

```
<div class="container mt-4">
  <!-- Paste the content of about.html here -->
</div>
```

Now we need to register the component in Angular routing. The file to update is the following:

src/app/app-routing.module.ts

Add an Import

```
import { AboutComponent } from '../themes/ilc-clarin/app/about/about.component';
```

Add a route

```
{
            path: 'about',
            component: AboutComponent,
            resolve: {
              breadcrumb: I18nBreadcrumbResolver
            },
            data: {
              title : 'about.title',
              breadcrumbKey: 'about',
              showBreadcrumbs: true
            }
          },
```

BEFORE THE ROUTE { path: “”, pathMatch: “full”, component:ThemedPageNotFoundComponent }**

To make the breadcrumb text visible, you need to modify the i18n JSON files.

src/assets/i18n/en.json5

Where you add:

"ilc-clarin.about.ilc4clarin-statement": "The ILC4CLARIN Repository",

“about.title” : “About”

“about.breadcrumbs”: “About”

And (finally) the header file **src/themes/ilc-clarin/app/header/header.component.html**

Where you add

```
<a href="{{ '/' + translateSlug('about') }}" class="lindat-dropdown-item">{{'ilc-clarin.about.ilc4clarin-statement' | translate}}</a>
```

Afterwards, the area must be cleaned up:

- Launch dspace-angular-claner.sh
- pm2 stop dspace-ui.json
- yarn build:prod
- pm2 start dspace-ui.json

For all the other components, you must follow the same procedure: create the component, register it in the routing module, and correctly copy the corresponding static HTML content into the newly created component's HTML file. The components to be added to the site are the following:

- FAQ
- Item Lifecycle
- Cite
- Privacy Policy
- Contract
- Term of service

Here there are the routing modules of each components:

```
{
            path: 'faq',
            component: FaqComponent,
            resolve: {
              breadcrumb: I18nBreadcrumbResolver
            },
            data: {
              title : 'faq.title',
              breadcrumbKey: 'faq',
              showBreadcrumbs: true
            }
          },
          {
            path: 'cite',
            component: CiteComponent,
            resolve: {
              breadcrumb: I18nBreadcrumbResolver
            },
            data: {
              title : 'cite.title',
              breadcrumbKey: 'cite',
              showBreadcrumbs: true
            }
          },
          {
            path: 'item-lifecycle',
            component: ItemSubmissionLifecycleComponent,
            resolve: {
              breadcrumb: I18nBreadcrumbResolver
```

```
      },
      data: {
        title : 'item-lifecycle.title',
        breadcrumbKey: 'item-lifecycle',
        showBreadcrumbs: true
      }
    },
    {
      path: 'privacy-policy',
      component: PrivacyPolicyComponent,
      resolve: {
        breadcrumb: I18nBreadcrumbResolver
      },
      data: {
        title : 'privacy-policy.title',
        breadcrumbKey: 'privacy-policy',
        showBreadcrumbs: true
      }
    },
    {
      path: 'contract',
      component: ContractComponent,
      resolve: {
        breadcrumb: I18nBreadcrumbResolver
      },
      data: {
        title : 'contract.title',
        breadcrumbKey: 'contract',
        showBreadcrumbs: true
      }
    },
    {
      path: 'terms-of-service',
      component: TermsOfServiceComponent,
      resolve: {
        breadcrumb: I18nBreadcrumbResolver
      },
      data: {
        title : 'terms-of-service.title',
        breadcrumbKey: 'terms-of-service',
        showBreadcrumbs: true
```

```
      }
    },
    {
      path: 'deposit',
      component: DepositComponent,
      resolve: {
        breadcrumb: I18nBreadcrumbResolver
      },
      data: {
        title : 'deposit.title',
        breadcrumbKey: 'deposit',
        showBreadcrumbs: true
      }
    },
    {
      path: 'metadata',
      component: MetadataComponent,
      resolve: {
        breadcrumb: I18nBreadcrumbResolver
      },
      data: {
        title : 'metadata.title',
        breadcrumbKey: 'metadata',
        showBreadcrumbs: true
      }
    },
```

4.7.1.2 Edit Header and Footer

To customize the **header** and **footer** of DSpace 7 and make them consistent with the previous DSpace 5 layout, follow these steps.

The header component is located at:

src/themes/ilc-clarin/app/header/header.component.html

Search the div with the class:

```
lindat-block lindat-block--clariah-theme-branding
```

First, upload the following image to /src/assets/images with the name clarin_it_logo.png.

The resulting HTML should look like this.

```
<div class="lindat-block lindat-block--clariah-theme-branding">
          <a href="https://www.clarin-it.it/it/" class="lindat-navbar-brand
lindat-d-flex lindat-align-items-center " aria-label="">
            <!-- <img src="./assets/images/ILC-CNR_for_CLARIN-IT_Logo.png"
width="auto" height="53" style="height: 53px !important;" alt="ILC4CLARIN logo"
class="" /> -->
             <img src="./assets/images/clarin_it_logo.png" alt="CLARIN-IT Logo"
class="" />
          </a>
        </div>
```

Find the div with the class:

```
lindat-collapse lindat-navbar-collapse
```

Comment all elements except “Home”, refactor “About” button like a lindat-nav-item. The output HTML be like:

```
<div class="lindat-block lindat-block--clariah-theme-main-menu">
              <ul class="lindat-nav lindat-navbar-nav">
                <!-- <li class="lindat-nav-item ">
                  <a href="/services/catalog" class="lindat-nav-
link">{{'navbar.community-list' | translate}}</a>
                </li> -->
                <li class="lindat-nav-item ">
                  <a routerLink="home" class="lindat-nav-
link">{{'navbar.repository' | translate}}</a>
                </li>
                <li class="lindat-nav-item ">
```

```
                <a routerLink="about" class="lindat-nav-link">{{'navbar.about'
| translate}}</a>
              </li>
              <!-- <li class="lindat-nav-item ">
                <a href="{{ '/' + getLangCodeIfCzech() + '#education' }}"
class="lindat-nav-link">{{'navbar.education' | translate}}</a>
              </li>
              <li class="lindat-nav-item ">
                <a href="{{ '/' + getLangCodeIfCzech() + '#projects' }}"
class="lindat-nav-link">{{'navbar.project' | translate}}</a>
              </li>
              <li class="lindat-nav-item ">
                <a href="{{ '/' + getLangCodeIfCzech() + '#tools' }}"
class="lindat-nav-link ">{{'navbar.tools' | translate}}</a>
              </li>
              <li class="lindat-nav-item ">
                <a href="{{ '/' + getLangCode() + '/' +
translateSlug('services') }}" class="lindat-nav-link ">{{'navbar.services' |
translate}}</a>
              </li>
              <li class="lindat-nav-item lindat-dropdown">
                <a href="{{ '/' + getLangCodeIfCzech() }}" class="lindat-nav-
link lindat-dropdown-toggle"
                   data-toggle="dropdown"
                   onclick="this.parentNode.querySelector('.lindat-dropdown-
toggle+div.lindat-dropdown-menu').classList.toggle('lindat-show'); return
false;">{{'navbar.about' | translate}}</a>
                <div class="lindat-dropdown-menu">
                  <a href="{{ '/' + translateSlug('partners') }}"
class="lindat-dropdown-item">{{'navbar.about.partners' | translate}}</a>
                  <a href="/assets/files/about.html" class="lindat-dropdown-
item">{{'navbar.about.mission-statement' | translate}}</a>
                  <a href="https://www.clarin.eu/" class="lindat-dropdown-
item">{{'navbar.about.clarin' | translate}}</a>
                  <a href="https://www.dariah.eu/" class="lindat-dropdown-
item">{{'navbar.about.dariah' | translate}}</a>
                  <a href="{{ '/' + translateSlug('integration') }}"
class="lindat-dropdown-item">{{'navbar.about.service-integrations' |
translate}}</a>
                  <a href="{{ '/' + translateSlug('partnership') }}"
class="lindat-dropdown-item">{{'navbar.about.project-partnership' |
```

```
translate}}</a>
                  </div>
                </li> -->
              </ul>
            </div>
```

Now we have to refix a last part, then search the div with class lindat-nav lindat-navbar-nav and comment out one section like (in order to remove the DARIAH logo and link):

```
<ul class="lindat-nav lindat-navbar-nav">
              <li class="lindat-nav-item" id="margin-filler"></li>
              <!-- <li class="lindat-nav-item  ">
                <a class="lindat-nav-link lindat-nav-link-dariah"
href="https://www.dariah.eu/"><img src="./assets/images/dariah-eu.png"
alt="DARIAH logo" /></a>
              </li> -->
              <li class="lindat-nav-item  ">
                <a class="lindat-nav-link lindat-nav-link-clarin"
href="https://www.clarin.eu/"><img src="./assets/images/clarin.png" alt="CLARIN
logo" /></a>
              </li>
            </ul>
```

For the footer, go to /src/app/footer and edit the footer.component.html like this:

```
<footer class="text-lg-start">
  <div class="lindat-common2 lindat-common-footer">
    <footer data-version="3.0.5" data-
build="05eff1186f12528f221a63b021c7b7dc81301429" style="grid-template-columns:
2fr 1fr 1fr !important;">
      <div id="about-lindat">
        <h4>Coordination, Partners, Funding</h4>
        <ul>
          <li><a href="http://www.ilc.cnr.it/">Institute for Computational
Linguistics "Antonio Zampolli" - Italian National Research Council</a></li>
```

```
          <li><a href="http://www.istruzione.it/">Italian Ministry of Education,
University and Research</a></li>
        </ul>
      </div>
      <div id="about-partners">
        <h4>Repository</h4>
        <ul>
          <li>
            <ul>
              <li><a routerLink="/home">Main Page</a></li>
              <li><a routerLink="/item-lifecycle">Submission Lifecycle</a></li>
            </ul>
          </li>
           <li>
            <ul>
              <li><a routerLink="/faq">FAQ</a></li>
              <li><a routerLink="/about">About and Policies</a></li>
              <li><a href="mailto:dspace-clarin-it-ilc-help@ilc.cnr.it">Help
Desk</a></li>
            </ul>
          </li>
        </ul>
      </div>
      <div id="about-website">
        <h4>About</h4>
        <ul>
          <li><a href="http://www.clarin.eu/">CLARIN EU</a></li>
          <li><a href="https://ilc4clarin.ilc.cnr.it/en/how-to-sign-up">How to
Sign Up</a></li>
        </ul>
      </div>
      <div id="badges-a">
        <a href="http://hdl.handle.net/11372/DOC-144"><img
src="./assets/images/footer/b-centre.png" style="    height: auto; width: 16rem;"
alt="CLARIN CENTRE B" /></a>
        <!-- <a href="https://www.clarin.eu/content/knowledge-centres"><img
src="./assets/images/footer/k-centre.png" alt="CLARIN CENTRE K"
style="filter:brightness(0.88)" /></a> -->
        <a href="https://www.coretrustseal.org/wp-content/uploads/2018/04/The-
ILC4CLARIN-Centre-at-the-Institute-for-Computational-Linguistics-.pdf"><img
src="./assets/images/footer/core-trust-seal-mono.png" style="width: 7rem; height:
```

```
auto;" alt="CoreTrustSeal Certification" /></a>
      </div>
      <div id="badges-b">
        <!-- <a href="https://twitter.com/lindatclariahcz">Follow us on Twitter
<img src="./assets/images/footer/twitter-circular.svg" alt="Link to Profile"
/></a> -->
        <a href="https://ilc4clarin.ilc.cnr.it/"><img src="./assets/images/ILC-
CNR_for_CLARIN-IT_Logo.png" alt="ILC4CLARIN Home Page" /></a>
      </div>
      <!-- <div id="ack-msmt">
        THE LINDAT/CLARIAH-CZ PROJECT (LM2018101; formerly LM2010013, LM2015071)
IS FULLY SUPPORTED BY THE MINISTRY OF EDUCATION, SPORTS AND YOUTH OF THE CZECH
REPUBLIC UNDER THE PROGRAMME LM OF "LARGE INFRASTRUCTURES"
      </div> -->
      <div id="ack-msmt">
        Copyright (c) 2025 ILC4CLARIN CNR. All rights reserved.
      </div>
      <div id="ack-freepik">Icons ©  Smashicons and Freepik from flaticon.com
licensed by <a href="https://creativecommons.org/licenses/by/3.0/">CC 3.0
BY</a></div>
      <div id="ack-ufal">website © 2024 by <a
href="https://ufal.mff.cuni.cz/">ÚFAL</a></div>
      <script type="text/javascript">
        //<![CDATA[

(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
          (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new
Date();a=s.createElement(o),

m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
        })(window,document,'script','//www.google-
analytics.com/analytics.js','ga');

        // main LINDAT/CLARIAH-CZ tracker
        ga('create', 'UA-27008245-2', 'cuni.cz');
        ga('send', 'pageview');
        //]]>
       </script>
    </footer>
  </div>
</footer>
```

Save, stop frontend and rebuild.

4.8 Test connection between backend and frontend

After deploying both the DSpace backend and frontend, it is important to verify that the two components communicate correctly through the REST API. This ensures that the frontend can retrieve data, authenticate users, and display content as expected.

To perform a basic connection test, run the following command from the root directory of the frontend project:

```
yarn test:rest
```

This command executes a predefined script that attempts to contact the backend REST API using the restApiBaseUrl defined in the environment configuration. It checks for proper connectivity and validates that the API endpoints respond correctly.

This is a good result:

```
yarn run v1.22.22
$ ts-node --project ./tsconfig.ts-node.json scripts/test-rest.ts
Building production app config
Overriding app config with /home/dev-clarin/dspace-angular/config/config.yml
Overriding app config with /home/dev-clarin/dspace-angular/config/config.production.yml
...Testing connection to REST API at https://dev-clarin.ilc.cnr.it/server/api...

RESPONSE: 200

Checking JSON returned for validity...
        "dspaceVersion" = DSpace 7.6.1
        "dspaceUI" = https://dev-clarin.ilc.cnr.it
        "dspaceServer" = https://dev-clarin.ilc.cnr.it/server
        "dspaceServer" property matches UI's "rest" config? true
        Does "/api" endpoint have HAL links ("_links" section)? true
Done in 5.37s.
```

If the connection test fails, be aware that several factors could be responsible:

- **Apache is not properly configured** or has not been started correctly. Ensure the virtual host, proxy rules, and enabled modules are correctly set up and that the service is running.
- **Tomcat is not running** or is experiencing issues. Confirm that the backend is deployed, reachable, and listening on the expected port.
- **SSL certificates are not correctly configured.** Make sure that your HTTPS setup (e.g., with Let's Encrypt or OpenSSL) is complete and matches the server configuration.
- **The /etc/hosts file is not correctly set.** Ensure that your machine resolves the backend domain (e.g., dev-clarin.ilc.cnr.it) to the correct local or internal IP address (e.g., 192.168.xxx.xxx).

Verifying each of these elements can help isolate and fix the issue.

4.9 Configure Shibboleth

To enable Shibboleth-based Single Sign-On (SSO), you must configure the shibboleth2.xml file located in the Shibboleth SP installation directory (typically /etc/shibboleth/shibboleth2.xml). This configuration defines your application's entityID, the metadata of the Identity Provider (IdP), the SSO bindings, and the protected resource rules.

In addition, you need to adjust the Apache configuration to ensure that requests to the Shibboleth handler are passed directly to the Shibboleth daemon and not forwarded by the reverse proxy. To achieve this, add the following directive to your default-ssl.conf (or equivalent virtual host configuration, before the <Location> directives):

```
ProxyPass /Shibboleth.sso !
```

After, you have to copy a particular file placed under /opt/shibboleth-sp/etc/shibboleth/, named apache24.conf. Copy it to the /etc/apache2 folder and include to apache2.conf.

After, restart Apache service.

Lastly, you have to configure your shibboleth2.xml file, please change these lines:

```
<SSO
                discoveryProtocol="SAMLDS"
                discoveryURL="https://discovery.clarin.eu/discojuice">
              SAML2 SAML1
            </SSO>

<Handler type="MetadataGenerator" Location="/Metadata" signing="false" template="/opt/shibboleth-
sp/etc/shibboleth/md_template.xml"/>
<MetadataProvider type="XML"
                          url="http://md.idem.garr.it/metadata/idem-metadata-sha256.xml"
                          backingFilePath="idem-metadata-sha256.xml" reloadInterval="7200"
legacyOrgNames="true">
            <MetadataFilter type="Signature"
                            certificate="/opt/shibboleth-sp/etc/shibboleth/idem-signer-20220121.pem"/>
        </MetadataProvider>

        <!-- CLARIN SPF - Metadata about all IdPs in participating Identity Federations -->

        <MetadataProvider type="XML"
              url="https://infra.clarin.eu/aai/prod_md_about_spf_idps.xml"
              backingFilePath="clarin-eu-spf-idps.xml" reloadInterval="7200" legacyOrgNames="true">
            <MetadataFilter type="Blacklist">
                <Exclude>
                        https://openidp.aco.net/saml
                </Exclude>
            </MetadataFilter>
        </MetadataProvider>

<MetadataProvider type="XML"
```

```
                        url="https://infra.clarin.eu/aai/prod_md_about_clarin_erics_idp.xml"
                        backingFilePath="homeless-clarin-eu-metadata.xml" reloadInterval="7200"
legacyOrgNames="true">
        </MetadataProvider>


        <!-- edugain IDP -->

        <MetadataProvider type="XML"
                        url="http://mds.edugain.org"
                        backingFilePath="edugain-metadata.xml" reloadInterval="7200"
legacyOrgNames="true">
        </MetadataProvider>

<CredentialResolver type="File" key="/etc/apache2/self_signed/dev-clarin.key"
certificate="/etc/apache2/self_signed/dev-clarin.crt"/>

<ApplicationOverride id="dev-clarin" entityID="https://dev-clarin.ilc.cnr.it/Shibboleth.sso/Metadata"
                                REMOTE_USER="eppn persistent-id targeted-id mail"
                                sessionHook="/clarin-sp-aaggregator/aa-statistics.php"

cipherSuites="DEFAULT:!EXP:!LOW:!aNULL:!eNULL:!DES:!IDEA:!SEED:!RC4:!3DES:!kRSA:!SSLv2:!SSLv3:!TLSv1:!T
LSv1.1">

                <Sessions lifetime="28800" timeout="3600" checkAddress="false" relayState="ss:mem"
handlerSSL="true"
                                        exportLocation="/GetAssertion" exportACL="127.0.0.1
192.168.93.26">
                <SSO entityID="https://idem-idp.ilc.cnr.it/idp/shibboleth">
                        SAML2 SAML1
                </SSO>
  </Sessions>

<AttributeExtractor type="XML" validate="true" reloadChanges="true" path="attribute-map.xml"/>

                <!-- Use a SAML query if no attributes are supplied during SSO. -->
                <AttributeResolver type="Query" subjectMatch="true"/>

                <!-- Default filtering policy for recognized attributes, lets other data pass. -->
                <AttributeFilter type="XML" validate="true" path="attribute-policy.xml"/>
                <!-- Default filtering policy for recognized attributes, lets other data pass. -->

                        <!-- Simple file-based resolver for using a single keypair. -->
                <!-- added new certificates 210121-->


                <CredentialResolver type="File" key="/etc/apache2/self_signed/dev-clarin.key"
certificate="/etc/apache2/self_signed/dev-clarin.crt"/>

<MetadataProvider
                    xmlns:xi="http://www.w3.org/2001/XInclude"
                    type="Chaining">
                    <!-- SAML metadata about all contracted identity federations' production IdPs. -->
                    <MetadataProvider
                        type="XML"
                        url="https://infra.clarin.eu/aai/prod_md_about_spf_idps.xml"
                        backingFilePath="prod_md_about_spf_idps.xml"
                        reloadInterval="7200" legacyOrgNames="true">
                        <MetadataFilter
                            xmlns="urn:mace:shibboleth:3.0:native:sp:config"
                            type="Blacklist">
                            <Exclude>https://openidp.aco.net/saml</Exclude>
                        </MetadataFilter>
                    </MetadataProvider>
                    <!-- SAML metadata about the CLARIN IdP. -->
                    <MetadataProvider
                        type="XML"
                        url="https://infra.clarin.eu/aai/prod_md_about_clarin_erics_idp.xml"
                        backingFilePath="prod_md_about_clarin_erics_idp.xml"
                        reloadInterval="7200" legacyOrgNames="true"/>
```

```
<MetadataProvider
                        type="XML"
                        url="http://mds.edugain.org"
                        backingFilePath="edugain-metadata-on-discofeed.xml"
                        reloadInterval="7200" legacyOrgNames="true"/>
                    <!-- metadata about the IDEM IdP. -->
                    <!-- sostituito il 31/1/2018
                    <MetadataProvider type="XML"
                        uri="http://www.garr.it/idem-metadata/idem-metadata-sha256.xml"
                        backingFilePath="idem-metadata-on-discofeed.xml" reloadInterval="7200"
legacyOrgNames="true">
                        <MetadataFilter type="Signature" certificate="/opt/shibboleth-
sp/etc/shibboleth/idem_signer_2019.pem"/>
                    </MetadataProvider>
                    -->
                    <MetadataProvider type="XML"
                        url="http://md.idem.garr.it/metadata/idem-metadata-sha256.xml"
                        backingFilePath="idem-metadata-sha256.xml" reloadInterval="7200"
legacyOrgNames="true">
                        <MetadataFilter type="Signature" certificate="/opt/shibboleth-
sp/etc/shibboleth/idem-signer-20220121.pem"/>
                    </MetadataProvider>

                </MetadataProvider>

        </ApplicationOverride>
```

You have to edit also the md_template.xml with this type of configuration:

```
<?xml version="1.0" encoding="utf-8"?>
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
  <Extensions xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
      <mdattr:EntityAttributes xmlns:mdattr="urn:oasis:names:tc:SAML:metadata:attribute">
        <saml:Attribute xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
Name="http://macedir.org/entity-category" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
          <saml:AttributeValue>http://www.geant.net/uri/dataprotection-code-of-
conduct/v1</saml:AttributeValue>
        </saml:Attribute>
      </mdattr:EntityAttributes>
  </Extensions>
  <md:SPSSODescriptor>
    <Extensions xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
      <mdui:UIInfo xmlns:mdui="urn:oasis:names:tc:SAML:metadata:ui">
         <mdui:DisplayName xml:lang="en">Digital Repository for the CLARIN Research Infrastructure
provided by ILC-CNR</mdui:DisplayName>
         <mdui:DisplayName xml:lang="it">Repository Digitale per la Infrastruttura di Ricerca CLARIN
erogato da ILC-CNR</mdui:DisplayName>
         <mdui:Description xml:lang="en">ILC-CNR for CLARIN-IT Consortium: Digital repository and
services on language research infrastructure</mdui:Description>
         <mdui:Description xml:lang="it">ILC-CNR per il consorzio CLARIN-IT: repository e servizi di
risorse linguistiche</mdui:Description>
         <mdui:InformationURL xml:lang="en">https://dspace-clarin-
it.ilc.cnr.it/repository/xmlui/page/about</mdui:InformationURL>
         <mdui:InformationURL xml:lang="it">https://dspace-clarin-
it.ilc.cnr.it/repository/xmlui/page/about</mdui:InformationURL>
         <mdui:Logo height="68" width="453" xml:lang="en">https://webmail.ilc.cnr.it/CLARIN-
IT_ILC_Logo_453x68.png</mdui:Logo>
         <mdui:Logo height="68" width="453" xml:lang="it">https://webmail.ilc.cnr.it/CLARIN-
IT_ILC_Logo_453x68.png</mdui:Logo>
         <mdui:Logo height="16" width="16"
xml:lang="en">https://webmail.ilc.cnr.it/ILC_Logo_IDEM_16x16.png</mdui:Logo>
```

```
        <mdui:Logo height="16" width="16"
xml:lang="it">https://webmail.ilc.cnr.it/ILC_Logo_IDEM_16x16.png</mdui:Logo>
        <mdui:Logo height="80" width="60"
xml:lang="en">https://webmail.ilc.cnr.it/ILC_Logo_IDEM_80x60.png</mdui:Logo>
        <mdui:Logo height="80" width="60"
xml:lang="it">https://webmail.ilc.cnr.it/ILC_Logo_IDEM_80x60.png</mdui:Logo>
       <mdui:PrivacyStatementURL xml:lang="en">https://dspace-clarin-
it.ilc.cnr.it/repository/xmlui/page/privacypolicy</mdui:PrivacyStatementURL>
        <mdui:PrivacyStatementURL xml:lang="it">https://dspace-clarin-
it.ilc.cnr.it/repository/xmlui/page/privacypolicy</mdui:PrivacyStatementURL>
      </mdui:UIInfo>
    </Extensions>
    <md:AttributeConsumingService index="1">
      <md:ServiceName xml:lang="en">Digital Repository for the CLARIN Research Infrastructure provided
by ILC-CNR</md:ServiceName>
      <md:ServiceName xml:lang="it">Repository Digitale per la Infrastruttura di Ricerca CLARIN erogato
da ILC-CNR</md:ServiceName>
      <md:ServiceDescription xml:lang="en">Digital Repository and services related to the CLARIN-IT
consortium under the CLARIN Research Infrastructure; focused in the field of Computational Linguistics,
it's provided by Institute for Computational Linguistics "Antonio Zampolli", National Research Council,
located in Pisa</md:ServiceDescription>
      <md:ServiceDescription xml:lang="it">Repository Digitale e servizi relativi al consorzio CLARIN-
IT sotto l'Infrastruttura di Ricerca CLARIN; focalizzato nel campo della Linguistica Computazionale, e'
fornito dall'Istituto di Linguistica Computazionale "Antonio Zampolli", Consiglio Nazionale delle
Ricerche, con sede a Pisa</md:ServiceDescription>
      <md:RequestedAttribute FriendlyName="eduPersonPrincipalName"
                             isRequired="true"
                             Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
      <md:RequestedAttribute FriendlyName="email"
                             isRequired="true"
                             Name="urn:oid:0.9.2342.19200300.100.1.3"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
      <md:RequestedAttribute FriendlyName="cn"
                             isRequired="false"
                             Name="urn:oid:2.5.4.3" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"/>
      <md:RequestedAttribute FriendlyName="schacHomeOrganization"
                             isRequired="false"
                             Name="urn:oid:1.3.6.1.4.1.25178.1.2.9"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
      <md:RequestedAttribute FriendlyName="organizationName"
                             isRequired="false"
                             Name="urn:oid:2.5.4.10" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri"/>
      <md:RequestedAttribute FriendlyName="displayName"
                             isRequired="true"
                             Name="urn:oid:2.16.840.1.113730.3.1.241"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
      <md:RequestedAttribute FriendlyName="eduPersonEntitlement"
                             isRequired="false"
                             Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
      <md:RequestedAttribute FriendlyName="eduPersonTargetedID"
                             isRequired="true"
                             Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
      <md:RequestedAttribute FriendlyName="eduPersonScopedAffiliation"
                             isRequired="true"
                             Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.9"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
    </md:AttributeConsumingService>
  </md:SPSSODescriptor>
  <md:Organization>
    <md:OrganizationName xml:lang="en">National Research Council (CNR)</md:OrganizationName>
    <md:OrganizationName xml:lang="it">Consiglio Nazionale delle Ricerche (CNR)</md:OrganizationName>
    <md:OrganizationDisplayName xml:lang="en">CNR Institute for Computational Linguistics "Antonio
Zampolli"</md:OrganizationDisplayName>
    <md:OrganizationDisplayName xml:lang="it">CNR Istituto di Linguistica Computazionale "Antonio
Zampolli"</md:OrganizationDisplayName>
    <md:OrganizationURL xml:lang="en">http://www.ilc.cnr.it/en</md:OrganizationURL>
```

```
    <md:OrganizationURL xml:lang="it">http://www.ilc.cnr.it</md:OrganizationURL>
  </md:Organization>
  <md:ContactPerson contactType="technical">
    <md:GivenName>Alessandro</md:GivenName>
    <md:SurName>Enea</md:SurName>
    <md:EmailAddress>mailto:alessandro.enea@ilc.cnr.it</md:EmailAddress>
  </md:ContactPerson>
  <md:ContactPerson contactType="technical">
    <md:GivenName>Riccardo</md:GivenName>
    <md:SurName>Del Gratta</md:SurName>
    <md:EmailAddress>mailto:riccardo.delgratta@ilc.cnr.it</md:EmailAddress>
  </md:ContactPerson>
  <md:ContactPerson contactType="support">
    <md:GivenName>Riccardo</md:GivenName>
    <md:SurName>Del Gratta</md:SurName>
    <md:EmailAddress>mailto:dspace-clarin-it-ilc-help@ilc.cnr.it</md:EmailAddress>
  </md:ContactPerson>
  <md:ContactPerson contactType="administrative">
    <md:GivenName>Riccardo</md:GivenName>
    <md:SurName>Del Gratta</md:SurName>
    <md:EmailAddress>mailto:dspace-clarin-it-ilc-help@ilc.cnr.it</md:EmailAddress>
  </md:ContactPerson></md:EntityDescriptor>
```

5. Test

This section focuses on verifying the correct installation and configuration of each component within the DSpace system. The goal is to ensure that the entire software stack—comprising the web server (Apache), application server (Tomcat), DSpace backend (REST API), and DSpace frontend (Angular)—is operational and properly integrated.

Each subsection provides instructions and tools to test individual services, confirm network accessibility, and validate the interaction between backend and frontend. Successful completion of these tests is a prerequisite for moving forward with production deployment or further customization.

5.1 Apache

The Apache HTTP Server plays a crucial role as the primary web server responsible for routing HTTP and HTTPS traffic to the appropriate backend services, including Tomcat (via proxy) and Shibboleth (for SSO). Ensuring that Apache is running correctly is a fundamental step before testing the higher-level application layers.

First, verify that Apache is installed and actively running:

```
sudo systemctl status apache2
```

A good practice is to check the configuration syntax using:

```
sudo apachectl configtest
```

If Apache is not running, start or restart the service:

```
sudo systemctl start apache2
# or to restart
sudo systemctl restart apache2
```

To enable Apache to start on boot:

```
sudo systemctl enable apache2
```

For any problems, you don't forget to check these paths:

- **Apache2 path**: /etc/apache2/
- **Apache2 main configuration file** : /etc/apache2/apache2.conf
- **Self signed certificates :** /etc/apache2/self_signed/
- **LetsEncrypt certificates:** /etc/letsencrypt/live/dev-clarin.ilc.cnr.it/
- **DSPACE configuration folder** : /etc/apache2/dspace-conf/
- **Shibboleth apache configuration file** : /etc/apache2/apache24.conf
- **Shibboleth DSPACE configuration file** : /etc/apache2/conf-available/shib-dev-clarin.conf
- **Apache2 Log folder** : /var/log/apache2/dev-clarin-*.log

In order to check the configuration, you can do a ping to https://dev-clarin.ilc.cnr.it/.

5.2 Tomcat

Ensure the Tomcat service is active:

```
sudo systemctl status tomcat
```

If inactive, start Tomcat:

```
sudo systemctl start tomcat
```

Try to go to ping the internal port using:

```
curl -v http://localhost:8080
```

For any problem, you can check the logs placed in /opt/tomcat/logs (viewable using sudo privileges).

5.3 Shibboleth

In order to test Shibboleth, you can see the logs placed in:

```
/opt/shibboleth-sp/var/log/shibboleth
```

Check if this URL retrieve an XML:

```
https://dev-clarin.ilc.cnr.it/Shibboleth.sso/Metadata
```

You should see an XML like this:

```
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
ID="_3236ca6526e6682876e441d03a70da65c5917a7b" entityID="https://dev-
clarin.ilc.cnr.it/Shibboleth.sso/Metadata">
  <Extensions xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
      <mdattr:EntityAttributes xmlns:mdattr="urn:oasis:names:tc:SAML:metadata:attribute">
        <saml:Attribute xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
Name="http://macedir.org/entity-category" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
          <saml:AttributeValue>http://www.geant.net/uri/dataprotection-code-of-
conduct/v1</saml:AttributeValue>
        </saml:Attribute>
      </mdattr:EntityAttributes>
    <alg:DigestMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2001/04/xmlenc#sha512"/>
    <alg:DigestMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha384"/>
    <alg:DigestMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
    <alg:DigestMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha224"/>
    <alg:DigestMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <alg:SigningMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha512"/>
    <alg:SigningMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha384"/>
    <alg:SigningMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256"/>
    <alg:SigningMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha224"/>
    <alg:SigningMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha512"/>
    <alg:SigningMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha384"/>
    <alg:SigningMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
    <alg:SigningMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2009/xmldsig11#dsa-sha256"/>
    <alg:SigningMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha1"/>
    <alg:SigningMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <alg:SigningMethod xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
  </Extensions>
  <md:SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol
urn:oasis:names:tc:SAML:1.1:protocol urn:oasis:names:tc:SAML:1.0:protocol">
    <Extensions xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
      <mdui:UIInfo xmlns:mdui="urn:oasis:names:tc:SAML:metadata:ui">
         <mdui:DisplayName xml:lang="en">Digital Repository for the CLARIN Research Infrastructure
provided by ILC-CNR</mdui:DisplayName>
         <mdui:DisplayName xml:lang="it">Repository Digitale per la Infrastruttura di Ricerca CLARIN
erogato da ILC-CNR</mdui:DisplayName>
         <mdui:Description xml:lang="en">ILC-CNR for CLARIN-IT Consortium: Digital repository and
services on language research infrastructure</mdui:Description>
         <mdui:Description xml:lang="it">ILC-CNR per il consorzio CLARIN-IT: repository e servizi di
risorse linguistiche</mdui:Description>
         <mdui:InformationURL xml:lang="en">https://dspace-clarin-
it.ilc.cnr.it/repository/xmlui/page/about</mdui:InformationURL>
```

```
        <mdui:InformationURL xml:lang="it">https://dspace-clarin-
it.ilc.cnr.it/repository/xmlui/page/about</mdui:InformationURL>
        <mdui:Logo height="68" width="453" xml:lang="en">https://webmail.ilc.cnr.it/CLARIN-
IT_ILC_Logo_453x68.png</mdui:Logo>
        <mdui:Logo height="68" width="453" xml:lang="it">https://webmail.ilc.cnr.it/CLARIN-
IT_ILC_Logo_453x68.png</mdui:Logo>
        <mdui:Logo height="16" width="16"
xml:lang="en">https://webmail.ilc.cnr.it/ILC_Logo_IDEM_16x16.png</mdui:Logo>
        <mdui:Logo height="16" width="16"
xml:lang="it">https://webmail.ilc.cnr.it/ILC_Logo_IDEM_16x16.png</mdui:Logo>
        <mdui:Logo height="80" width="60"
xml:lang="en">https://webmail.ilc.cnr.it/ILC_Logo_IDEM_80x60.png</mdui:Logo>
        <mdui:Logo height="80" width="60"
xml:lang="it">https://webmail.ilc.cnr.it/ILC_Logo_IDEM_80x60.png</mdui:Logo>
       <mdui:PrivacyStatementURL xml:lang="en">https://dspace-clarin-
it.ilc.cnr.it/repository/xmlui/page/privacypolicy</mdui:PrivacyStatementURL>
        <mdui:PrivacyStatementURL xml:lang="it">https://dspace-clarin-
it.ilc.cnr.it/repository/xmlui/page/privacypolicy</mdui:PrivacyStatementURL>
      </mdui:UIInfo>
      <init:RequestInitiator xmlns:init="urn:oasis:names:tc:SAML:profiles:SSO:request-init"
Binding="urn:oasis:names:tc:SAML:profiles:SSO:request-init" Location="https://dev-
clarin.ilc.cnr.it/Shibboleth.sso/Login"/>
      <idpdisc:DiscoveryResponse xmlns:idpdisc="urn:oasis:names:tc:SAML:profiles:SSO:idp-discovery-
protocol" Binding="urn:oasis:names:tc:SAML:profiles:SSO:idp-discovery-protocol" Location="https://dev-
clarin.ilc.cnr.it/Shibboleth.sso/Login" index="1"/>
    </Extensions>
    <md:KeyDescriptor>
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:KeyName>Istituto di Linguistica Computazionale "A. Zampolli"</ds:KeyName>
        <ds:X509Data>
          <ds:X509SubjectName>emailAddress=michele.mallia@ilc.cnr.it,CN=Istituto di Linguistica
Computazionale \"A. Zampolli\",O=ILC-CNR,L=Pisa,ST=PI,C=IT</ds:X509SubjectName>
          <ds:X509Certificate>MIIGKzC...
</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
      <md:EncryptionMethod Algorithm="http://www.w3.org/2009/xmlenc11#aes128-gcm"/>
      <md:EncryptionMethod Algorithm="http://www.w3.org/2009/xmlenc11#aes192-gcm"/>
      <md:EncryptionMethod Algorithm="http://www.w3.org/2009/xmlenc11#aes256-gcm"/>
      <md:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
      <md:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes192-cbc"/>
      <md:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc"/>
      <md:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc"/>
      <md:EncryptionMethod Algorithm="http://www.w3.org/2009/xmlenc11#rsa-oaep"/>
      <md:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p"/>
    </md:KeyDescriptor>
    <md:ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="https://dev-clarin.ilc.cnr.it/Shibboleth.sso/Artifact/SOAP" index="1"/>
    <md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP" Location="https://dev-
clarin.ilc.cnr.it/Shibboleth.sso/SLO/SOAP"/>
    <md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://dev-clarin.ilc.cnr.it/Shibboleth.sso/SLO/Redirect"/>
    <md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://dev-clarin.ilc.cnr.it/Shibboleth.sso/SLO/POST"/>
    <md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"
Location="https://dev-clarin.ilc.cnr.it/Shibboleth.sso/SLO/Artifact"/>
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://dev-clarin.ilc.cnr.it/Shibboleth.sso/SAML2/POST" index="1"/>
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign"
Location="https://dev-clarin.ilc.cnr.it/Shibboleth.sso/SAML2/POST-SimpleSign" index="2"/>
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"
Location="https://dev-clarin.ilc.cnr.it/Shibboleth.sso/SAML2/Artifact" index="3"/>
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:PAOS"
Location="https://dev-clarin.ilc.cnr.it/Shibboleth.sso/SAML2/ECP" index="4"/>
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-post"
Location="https://dev-clarin.ilc.cnr.it/Shibboleth.sso/SAML/POST" index="5"/>
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:1.0:profiles:artifact-01"
Location="https://dev-clarin.ilc.cnr.it/Shibboleth.sso/SAML/Artifact" index="6"/>
    <md:AttributeConsumingService index="1">
      <md:ServiceName xml:lang="en">Digital Repository for the CLARIN Research Infrastructure provided
by ILC-CNR</md:ServiceName>
```

```
      <md:ServiceName xml:lang="it">Repository Digitale per la Infrastruttura di Ricerca CLARIN erogato
da ILC-CNR</md:ServiceName>
      <md:ServiceDescription xml:lang="en">Digital Repository and services related to the CLARIN-IT
consortium under the CLARIN Research Infrastructure; focused in the field of Computational Linguistics,
it's provided by Institute for Computational Linguistics "Antonio Zampolli", National Research Council,
located in Pisa</md:ServiceDescription>
      <md:ServiceDescription xml:lang="it">Repository Digitale e servizi relativi al consorzio CLARIN-
IT sotto l'Infrastruttura di Ricerca CLARIN; focalizzato nel campo della Linguistica Computazionale, e'
fornito dall'Istituto di Linguistica Computazionale "Antonio Zampolli", Consiglio Nazionale delle
Ricerche, con sede a Pisa</md:ServiceDescription>
      <md:RequestedAttribute FriendlyName="eduPersonPrincipalName"
Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
isRequired="true"/>
      <md:RequestedAttribute FriendlyName="email" Name="urn:oid:0.9.2342.19200300.100.1.3"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri" isRequired="true"/>
      <md:RequestedAttribute FriendlyName="cn" Name="urn:oid:2.5.4.3"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri" isRequired="false"/>
      <md:RequestedAttribute FriendlyName="schacHomeOrganization"
Name="urn:oid:1.3.6.1.4.1.25178.1.2.9" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
isRequired="false"/>
      <md:RequestedAttribute FriendlyName="organizationName" Name="urn:oid:2.5.4.10"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri" isRequired="false"/>
      <md:RequestedAttribute FriendlyName="displayName" Name="urn:oid:2.16.840.1.113730.3.1.241"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri" isRequired="true"/>
      <md:RequestedAttribute FriendlyName="eduPersonEntitlement"
Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
isRequired="false"/>
      <md:RequestedAttribute FriendlyName="eduPersonTargetedID"
Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
isRequired="true"/>
      <md:RequestedAttribute FriendlyName="eduPersonScopedAffiliation"
Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.9" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
isRequired="true"/>
    </md:AttributeConsumingService></md:SPSSODescriptor>
  <md:Organization>
    <md:OrganizationName xml:lang="en">National Research Council (CNR)</md:OrganizationName>
    <md:OrganizationName xml:lang="it">Consiglio Nazionale delle Ricerche (CNR)</md:OrganizationName>
    <md:OrganizationDisplayName xml:lang="en">CNR Institute for Computational Linguistics "Antonio
Zampolli"</md:OrganizationDisplayName>
    <md:OrganizationDisplayName xml:lang="it">CNR Istituto di Linguistica Computazionale "Antonio
Zampolli"</md:OrganizationDisplayName>
    <md:OrganizationURL xml:lang="en">http://www.ilc.cnr.it/en</md:OrganizationURL>
    <md:OrganizationURL xml:lang="it">http://www.ilc.cnr.it</md:OrganizationURL>
  </md:Organization>
  <md:ContactPerson contactType="technical">
    <md:GivenName>Alessandro</md:GivenName>
    <md:SurName>Enea</md:SurName>
    <md:EmailAddress>mailto:alessandro.enea@ilc.cnr.it</md:EmailAddress>
  </md:ContactPerson>
  <md:ContactPerson contactType="technical">
    <md:GivenName>Riccardo</md:GivenName>
    <md:SurName>Del Gratta</md:SurName>
    <md:EmailAddress>mailto:riccardo.delgratta@ilc.cnr.it</md:EmailAddress>
  </md:ContactPerson>
  <md:ContactPerson contactType="support">
    <md:GivenName>Riccardo</md:GivenName>
    <md:SurName>Del Gratta</md:SurName>
    <md:EmailAddress>mailto:dspace-clarin-it-ilc-help@ilc.cnr.it</md:EmailAddress>
  </md:ContactPerson>
  <md:ContactPerson contactType="administrative">
    <md:GivenName>Riccardo</md:GivenName>
    <md:SurName>Del Gratta</md:SurName>
    <md:EmailAddress>mailto:dspace-clarin-it-ilc-help@ilc.cnr.it</md:EmailAddress>
  </md:ContactPerson>
</md:EntityDescriptor>
```

5.4 DSPACE Backend

To check if the backend is running, please go to:

```
https://dev-clarin.ilc.cnr.it/server
```

5.5 DSPACE Frontend

To check if the backend is running, please go to:

```
https://dev-clarin.ilc.cnr.it/
```

6. Migration procedure

This section outlines the steps to migrate an existing DSpace installation to a new server while preserving all data, configurations, and functionality. The process involves transferring the database, assetstore, and configurations, followed by validation and cleanup.

Before starting, ensure a complete backup of the following components from the **source server**:

```
pg_dump -U dspace -Fc dspace > dspace_backup.dump
```

Copy the entire [dspace]/assetstore directory (contains bitstreams):

```
tar -czvf assetstore_backup.tar.gz /opt/dspace/assetstore
```

Archive custom configurations (e.g., local.cfg, dspace.cfg, themes, SOLR cores).

```
tar -czvf config_backup.tar.gz /dspace/config
```

Now prepare the backup of the old database.

6.1 DSPACE 5 Database Backup

This procedure describes the steps required to export, compress, transfer, and extract the databases used by the DSpace system in preparation for the migration to the DSpace 7 instance.

Log in to the *dev-clarin2* machine as the **dspace** user and export the two DSpace-related databases:

- **ilcforclarin-db**
- **Ilcforclarin-db-utils**

```
pg_dump -U dspace ilcforclarin-db > ilcforclarin-db.dump
pg_dump -U dspace ilcforclarin-db-utils > ilcforclarin-db-utils.dump
```

After exporting the databases, compress both dump files using **tar**:

```
tar -czvf backup_databases.tar.gz ilcforclarin-db.dump ilcforclarin-db-utils.dump
```

This creates a single compressed archive containing both database dumps.

Copy the compressed archive from *dev-clarin2* (user **dev-clarin2**) to the *dev-clarin* machine (user **dev-clarin**) via **scp**, placing it in the designated backup directory:

```
scp backup_databases.tar.gz dev-clarin@dev-clarin:/var/dspace7_mount_disk/backup_databases_XX_XX_XXXX/
```

Replace the placeholder with the actual backup date.

Once the archive has been transferred to *dev-clarin*, navigate to the target directory and extract its contents using:

```
cd /var/dspace7_mount_disk/backup_databases_XX_XX_XXXX/
tar -xzvf backup_databases.tar.gz
```

After extraction, both .dump files will be available for the subsequent database import steps. All permissions must be assigned to tomcat user (to consent the web application to access these files).

6.2 Stopping Shibboleth, Apache, Tomcat, and DSpace Frontend Services

Before starting the migration process, all running services related to authentication, web access, backend processing, and the DSpace Angular UI must be safely stopped. This ensures data consistency and prevents conflicts during the migration procedure.

6.2.1 Stopping Shibboleth Service

List Shibboleth-related processes:

```
ps aux | grep "shib"
```

Identify the corresponding **PID**.

Terminate the process (as sudo user) using:

```
kill -9 <PID>
```

6.2.2 Stopping Apache and Tomcat

Both Apache HTTPD and Tomcat can be stopped using standard systemd commands:

```
sudo systemctl stop apache2
sudo systemctl stop tomcat
```

This ensures that all web traffic and backend servlet operations are safely interrupted before migration.

6.2.3 Stopping the DSpace Frontend (Angular UI)

The DSpace 7 Angular UI is managed through **PM2**. The service must be stopped using the non-root user **dev-clarin**.

Navigate to the DSpace Angular directory:

```
cd /home/dev-clarin/dspace-angular/
```

Stop the running UI instance:

```
pm2 stop dspace-ui.json
```

This halts the DSpace frontend application while preserving its PM2 configuration.

6.3 Deleting Old DSpace 7 Databases

Before restoring the new database dumps, any existing DSpace 7 databases and related extensions must be removed. This ensures that the migration starts from a clean and consistent PostgreSQL environment.

Connect to the server with the **postgres** user (or use sudo when required) and drop the old databases used by the previous DSpace 7 installation:

```
dropdb --username=postgres ilcforclarin-db
dropdb --username=postgres ilcforclarin-db-utils
```

These commands permanently remove the databases, freeing the names for the new restored instances.

If extensions such as **pgcrypto** were manually enabled in previous setups, they must also be removed before reinitializing the database.

To drop the extension, execute:

```
sudo -u postgres psql -d dspace -c "DROP EXTENSION pgcrypto;"
```

This command removes the extension from the target database, preventing conflicts during the new database import procedure.

6.4 Importing the New DSpace 7 Databases

Once the old databases have been removed, the new DSpace 7 databases can be recreated and populated using the previously generated backup files. This section describes how to recreate the database structures, import the data, restore required extensions, and apply post-import adjustments.

Using the PostgreSQL superuser (**postgres**), create the two databases required by the DSpace installation. Both databases must be owned by the **dspace** user and encoded in **UNICODE**.

```
createdb --username=postgres --owner=dspace --encoding=UNICODE ilcforclarin-db
createdb --username=postgres --owner=dspace --encoding=UNICODE ilcforclarin-db-utils
```

This step initializes two empty databases ready to receive the imported data.

After recreating the databases, import the SQL dump files previously extracted into the designated backup directory:

```
psql ilcforclarin-db < /var/dspace7_mount_disk/backup_databases_XX_XX_XXXX/ilcforclarin-db
psql ilcforclarin-db-utils < /var/dspace7_mount_disk/backup_databases_XX_XX_XXXX/ilcforclarin-db-utils
```

These commands populate the newly created databases with all schema definitions, data, and stored metadata from the source instance.

Some DSpace functionalities rely on the **pgcrypto** extension (e.g., token generation or hashing mechanisms).
After importing the databases, recreate the extension using:

```
sudo -u postgres psql -d dspace -c "CREATE EXTENSION pgcrypto;"
```

This ensures that cryptographic functions required by DSpace are available and properly initialized.

After the import is complete, apply the following update to correct the reference to the eperson_id within license definitions:

```
update license_definition set eperson_id = 10 where eperson_id = 1;
```

Reactivate Apache (with Shibboleth) and Tomcat services.

After importing the databases and recreating the required PostgreSQL extensions, the DSpace 7 database schema must be initialized using the built-in DSpace migration tool.

From the DSpace 7 installation directory:

```
cd /dspace7/bin
./dspace database migrate force
```

The force flag ensures that all required local type conversions and schema adjustments are executed, even if the system believes the database is already up to date.
This step is mandatory to bring the imported database in full alignment with the DSpace 7 schema.

Once the database schema has been initialized, create the administrator user needed to access and configure the DSpace backend:

```
./dspace create-administrator
```

This interactive command prompts for email, password, first name, and last name, and registers a fully privileged administrator account in the restored DSpace 7 environment.

6.5 Downloading the DSpace Migration Tool

The DSpace Migration Tool is required to perform the upgrade of database schemas, metadata structures, and internal objects when moving from the old DSpace version to DSpace 7.
This section describes how to obtain the tool package from the official DSpace repository and prepare it for execution.

The migration tool is available at the following GitHub repository: https://github.com/ufal/dspace-migrate

This repository contains the migration scripts, documentation, and utilities needed to perform the upgrade process.

On the *dev-clarin* machine, clone the repository using git (under opt folder):

```
git clone https://github.com/ufal/dspace-migrate.git
```

Enter in the folder, then cd into “src” folder and find the file named project_settings.py.

Configure project_settings.py with:

```
import os
from datetime import datetime
_this_dir = os.path.dirname(os.path.abspath(__file__))
ts = datetime.now().strftime("%Y_%m_%d__%H.%M.%S")

settings = {
    "log_file": os.path.join(_this_dir, "../__logs", f"{ts}.txt"),
```

```
    "resume_dir": "__temp/resume/",

    "backend": {
        "endpoint": "https://dev-clarin.ilc.cnr.it/server/api/",
        "user": "xxxxxx", anonymized for security reason
        "password": "xxxxxxx",  anonymized for security reason
        "authentication": True,
        "testing": True,
    },

    "ignore": {
        "missing-icons": ["PUB", "RES", "ReD", "Inf"],
        "epersons": [
            # ignore - empty person
            198
        ],
    },

    "db_dspace_7": {
        # CLARIN-DSpace 7 database
        "name": "dspace",
        "host": "localhost",
        # careful - NON standard port
        "port": 5432,
        "user": "dspace",
        "password": "xxxxxx",  your password of dspace7 db
    },

    "db_dspace_5": {
        "name": "ilcforclarin-db",
        "host": "localhost",
        "user": "dspace",
        "password": "xxxxxxxxxx",  your password of dspace5 db
        "port": 5432,
    },

    "db_utilities_5": {
        "name": "ilcforclarin-db-utils",
        "host": "localhost",
        "user": "dspace",
        "password": "xxxxxxxxx",  your password of dspace5 db
        "port": 5432,
    },

    "input": {
        "tempdbexport": os.path.join(_this_dir, "../input/tempdbexport"),
        "icondir": os.path.join(_this_dir, "../input/icon"),
    },

    "licenses": {
        "to_replace_def_url": "https://dspace-clarin-it.ilc.cnr.it/repository/xmlui/page/",
        #
        "replace_with_def_url": "https://dev-clarin.ilc.cnr.it/static/",
    }
}
```

Ensure services are running:

- PostgreSQL
- Solr
- DSpace backend

Launch migration:

```
python repo_import.py --assetstore=/opt/dspace/assetstore
```

After this, restart Tomcat and Frontend services.

6.6 Disabling Direct Download from S3

To ensure that all file downloads are routed through the DSpace application layer—rather than served directly from the S3 storage backend—the direct download option must be disabled in the configuration.

Navigate to the DSpace configuration folder:

```
cd /dspace/config
```

Edit the clarin-dspace.cfg file using vim editor.

Search for the following parameter:

```
s3.download.direct.enabled
```

Set the parameter to:

```
s3.download.direct.enabled = false
```

After modifying the configuration, restart the required services: Tomcat and DSPACE frontend.

6.7 Import OAI indexes

To ensure the correct exposure of metadata via the OAI-PMH protocol, DSpace provides a command-line utility to generate or rebuild the internal OAI index. This operation is typically required after bulk imports, data migrations, or significant metadata updates.

The OAI index can be imported using the following command (via sudo):

```
/dspace/bin/dspace oai import -c
```

The -c (clean) option removes the existing OAI index before rebuilding it, ensuring consistency and preventing outdated or duplicated records.

7. Start Handle Server

To enable persistent identifier resolution via the Handle System, the DSpace Handle Server was configured and started using the following procedure.

First, the default Handle Server configuration files were generated by running the following command:

```
/dspace/bin/make-handle-config
```

This command initializes the Handle Server directory structure and creates the required configuration templates.

When migrating an existing DSpace installation or reusing a previously configured Handle Server, the sitebundle.zip file must be copied from the former machine and restored on the new system.

Then, navigate to the Handle Server configuration directory:

```
cd /dspace/handle-server/
```

Copy the sitebundle.zip file from the previous machine to the new server, placing it in the Handle Server directory and extract it.

Open the main configuration file for editing:

```
vi config.dct
```

Update the file by inserting the following configuration values:

```
{
  "hdl_http_config" = {
    "bind_address" = "192.168.xx.xx" → insert real IP of your machine
    "num_threads" = "15"
    "bind_port" = "8010"
    "log_accesses" = "no"
  }

  "server_type" = "server"

  "hdl_udp_config" = {
    "bind_address" = "192.168.xx.xx"
    "num_threads" = "15"
    "bind_port" = "2641"
    "log_accesses" = "no"
  }

  "hdl_tcp_config" = {
    "bind_address" = "192.168.xx.xx"
    "num_threads" = "15"
    "bind_port" = "2641"
    "log_accesses" = "no"
  }

  "log_save_config" = {
    "log_save_directory" = "logs"
    "log_save_interval" = "Monthly"
  }

  "no_udp_resolution" = "no"
```

```
  "interfaces" = (
    "hdl_udp"
    "hdl_tcp"
    "hdl_http"
  )

  "server_config" = {
    "storage_type" = "CUSTOM"
    "storage_class" = "org.dspace.handle.HandlePlugin"
    "enable_txn_queue" = "no"

    "server_admins" = (
      "300:0.NA/20.500.11752" → this is your handle obtained previously by handle.net
    )

    "replication_admins" = (
      "300:0.NA/20.500.11752"
    )

    "max_session_time" = "86400000"
    "this_server_id" = "1"
    "max_auth_time" = "60000"
    "server_admin_full_access" = "yes"
    "case_sensitive" = "no"

    "auto_homed_prefixes" = (
      "0.NA/20.500.11752"
    )

    "case_sensitive" = "yes"
  }
}
```

This configuration defines the network interfaces, ports, administrative handles, and storage backend required for Handle resolution within DSpace.

Save the file and exit the editor once the configuration has been completed.

Due to compatibility requirements of the Handle Server and the others component builded, the Java runtime must be compatible with the java version of the all compiled software (Java 11).

Edit the executable script used to start the Handle Server (typically named start-handle-server) and explicitly set the Java binary as follows:

```
JAVA=/usr/lib/jvm/java-11-openjdk-amd64/bin/java
```

This ensures that the Handle Server is executed using Java 11 instead of the system default Java version.

Start the Handle Server using the provided DSpace script:

```
/dspace/bin/start-handle-server
```

To confirm that the Handle Server is working as expected, test handle resolution by accessing the following URL in a web browser:

```
http://hdl.handle.net/20.500.11752
```

If the configuration is correct, the handle should successfully resolve and give you a page with the handle information about your institution or instance registered with the handle.net.

8. Change item submission

By default, DSpace enforces file upload as a mandatory step during item submission. However, it is possible to customize the submission workflow in order to make file upload optional for specific collections, while keeping it mandatory for others.

This customization is achieved by defining a new upload configuration and associating it with a dedicated submission process.

Edit the file:

```
/dspace/config/spring/api/access-conditions.xml
```

Below the existing uploadConfigurationDefault bean, define a new UploadConfiguration bean identical to the default one, except for the required property, which must be set to false.

```
<!-- Upload configuration with non-mandatory file upload -->
<bean id="uploadConfigurationWithoutMandatory"
      class="org.dspace.submit.model.UploadConfiguration">
    <property name="name" value="upload" />
    <property name="metadata" value="bitstream-metadata" />
    <property name="required" value="false" />
    <property name="options">
        <list>
            <ref bean="openAccess" />
            <ref bean="lease" />
            <ref bean="embargoed" />
            <ref bean="administrator" />
        </list>
    </property>
</bean>
```

Below the existing uploadConfigurationService bean in the same file, define a new service that maps a custom key to the optional upload configuration:

```
<!-- UploadConfigurationService for optional file upload -->
<bean id="uploadConfigurationServiceWithoutMandatory"
      class="org.dspace.submit.model.UploadConfigurationService">
    <property name="map">
        <map>
            <entry key="uploadOptional"
                   value-ref="uploadConfigurationWithoutMandatory" />
        </map>
    </property>
</bean>
```

This service will later be referenced by the submission step to control upload behavior.

Edit the submission workflow configuration file:

```
/dspace/config/item-submission.xml
```

Below the existing upload step definition, create a new step that uses the optional upload configuration:

```
<!-- Upload step with optional file upload -->
<step-definition id="uploadOptional">
    <heading>submit.progressbar.upload</heading>
    <processing-class>
        org.dspace.app.rest.submit.step.UploadStep
    </processing-class>
    <type>upload</type>
</step-definition>
```

Finally, define a new submission process that replaces the default upload step with the newly created uploadOptional step.

Example of a submission process **with mandatory file upload**:

```
<submission-process name="traditional">
    <step id="collection" />
    <step id="traditionalpageone" />
    <step id="traditionalpagetwo" />
    <step id="upload" />
    <step id="license" />
</submission-process>
```

Example of a submission process **with optional file upload**:

```
<submission-process name="traditionalUploadOptional">
    <step id="collection" />
    <step id="traditionalpageone" />
    <step id="traditionalpagetwo" />
    <step id="uploadOptional" />
    <step id="license" />
</submission-process>
```

Any collection configured to use the traditionalUploadOptional submission process will no longer require file upload during item submission.
Collections that continue to use the standard traditional submission process will still enforce mandatory file upload.

This approach allows fine-grained control over submission requirements on a per-collection basis, without affecting the global DSpace configuration.

9. Change default license

To ensure consistency with the previous DSpace installation, the default license text was updated to match the license used in the legacy system.

Navigate to the DSpace configuration directory and open the file `default.license`.

Add this content:

```
NON-EXCLUSIVE DISTRIBUTION LICENSE

By signing and submitting this license, you (the author(s) or copyright owner)
grants to Institute for Computational Linguistics "A. Zampolli" (ILC) the
non-exclusive right to reproduce, translate (as defined below), and/or distribute
your submission (including the abstract) worldwide in print and electronic format
and in any medium, including but not limited to audio or video.

You agree that ILC may, without changing the content, translate the
submission to any medium or format for the purpose of preservation.

You also agree that ILC may keep more than one copy of this submission for
purposes of security, back-up and preservation.

You represent that the submission is your original work, and that you have
the right to grant the rights contained in this license. You also represent
that your submission does not, to the best of your knowledge, infringe upon
anyone's copyright.

If the submission contains material for which you do not hold copyright,
you represent that you have obtained the unrestricted permission of the
copyright owner to grant ILC the rights required by this license, and that
such third-party owned material is clearly identified and acknowledged
within the text or content of the submission.

IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED
BY AN AGENCY OR ORGANIZATION OTHER THAN ILC, YOU REPRESENT THAT YOU HAVE
FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH
CONTRACT OR AGREEMENT.

ILC will clearly identify your name(s) as the author(s) or owner(s) of the
submission, and will not make any alteration, other than as allowed by this
license, to your submission.
```

Save the file and reload tomcat.

10. Add Matomo JS Script for statistics

To enable usage statistics collection, Matomo analytics was integrated into the DSpace frontend. A dedicated Matomo server had been previously set up by the system administrator, and a specific user account was created and associated with the DSpace instance. Together with the account, a JavaScript tracking snippet was provided to be embedded into the DSpace web application.

The Matomo JavaScript code must be added to the main HTML entry point of the DSpace Angular frontend editing the following file:

```
/home/dev-clarin/dspace-angular/src/index.html
```

Insert the provided Matomo tracking script inside the file (typically within the <head> section):

```
<!-- Matomo -->
<script>
  var _paq = window._paq = window._paq || [];
  /* tracker methods like "setCustomDimension" should be called before "trackPageView" */
  _paq.push(['trackPageView']);
  _paq.push(['enableLinkTracking']);
  (function() {
    var u="https://stats.ilc.cnr.it/";
    _paq.push(['setTrackerUrl', u+'matomo.php']);
    _paq.push(['setSiteId', '12']);
    var d=document, g=d.createElement('script'), s=d.getElementsByTagName('script')[0];
    g.async=true;
    g.src=u+'matomo.js';
    s.parentNode.insertBefore(g,s);
  })();
</script>
<!-- End Matomo Code -->
```

Save the file once the script has been added.

After modifying the frontend source code, the Angular application must be rebuilt and restarted. All commands below are executed as a **non-root user**.

Stop the running frontend instance using `pm2 stop dspace-ui.json` from the root path of the dpsace-angular folder.

Build the production version of the frontend with `yarn build:prod` and relaunch the frontend application with `pm2 start dspace-ui.json`.

After the frontend is restarted, Matomo will begin collecting usage statistics for the DSpace web interface according to the configuration defined on the Matomo server.

The Matomo script doesn't work when you have ad-blocker extensions installed on your browser.