Research and Innovation Action

Social Sciences & Humanities Open Cloud

Project Number: 823782 Start Date of Project: 01/01/2019 Duration: 40 months

Deliverable 5.5 'Archive in a Box' repository software and proof of concept of centralised installation in the cloud

Dissemination Level	PU
Due Date of Deliverable	28/02/2022, (M38)
Actual Submission Date	31/03/2022, (M39)
Work Package	WP5 Innovation in data access
Task	5.2 Hosting and sharing data repositories
Type	Other
Approval Status	Approved by EC - 27 April 2022
Version	V1.0
Number of Pages	p.1 – p.33

Abstract:
This document reports about the 'Archive in a Box' and the proof of concept of a cloud installation. Furthermore, it reports about additional functionalities developed for the Dataverse software by the SSHOC task 5.2. Hosting and sharing data repositories.

The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided "as is" without guarantee or warranty of any kind, express or implied, including but not limited to the fitness of the information for a particular purpose. The user thereof uses the information at his/ her sole risk and liability.

SSHOC, "Social Sciences and Humanities Open Cloud", has received funding from the EU Horizon 2020 Research and Innovation Programme (2014-2020); H2020-INFRAEOSC-04-2018, under the agreement No. 823782

History

Version	Date	Reason	Revised by
0.0	31/01/2022	Initial draft	Marion Wittenberg
0.1	07/02/2022	First version	Marion Wittenberg
0.2	11/02/2022	Version ready for internal review	Marion Wittenberg
0.3	14/02/2022	Version ready for peer review	Marion Wittenberg
0.4	21/02/2022	Address peer review comments	Marion Wittenberg / Vyacheslav Tykhonov
1.0	21/02/2022	Version sent to WP lead and coordinator	Marion Wittenberg

Author List

Organisation	Name	Contact Information
KNAW/DANS (CESSDA)	Marion Wittenberg	marion.wittenberg@dans.knaw.nl
KNAW/DANS (CESSDA)	Vyacheslav Tykhonov	vyacheslav.tykhonov@dans.knaw.nl
KNAW/DANS (CESSDA)	Eko Indarto	eko.indarto@dans.knaw.nl
KNAW/DANS (CESSDA)	Wilko Steinhoff	wilko.steinhoff@dans.knaw.nl
KNAW/DANS (CESSDA)	Laura Huis in ‘t Veld	laura.huisintveld@dans.knaw.nl
AUSSDA/University of Vienna (CESSDA)	Stefan Kasberger	stefan.kasberger@univie.ac.at
University of Tromsø (CLARIN)	Philipp Conzett	philipp.conzett@uit.no
CNR (E-HRIS)	Cesare Concordia	cesare.concordia@isti.cnr.it
Universität Göttingen (DARIAH)	Peter Kiraly	peter.kiraly@gwdg.de
PSNC (DARIAH)	Tomasz Parkoła	tparkola@man.poznan.pl

Executive Summary

Within task 5.2 (Hosting and sharing data repositories) of the SSHOC project, repository software is being developed based on Dataverse[1], for the sharing and publication of research data within the Social Science and Humanities (SSH) domain. Dataverse is open-source research data repository software developed by the Institute for Quantitative Social Science (IQSS), Harvard University. This document describes the work done by task 5.2, for the development of 'Archive in a Box' repository software and proof of concept of centralised installation in the cloud.

The 'Archive in a Box' makes the installation of Dataverse repository software easier for institutes with a lack of technical staff. This document describes the advantages of such a package.

Additionally, task 5.2 worked on a proof of concept of a centralised cloud installation of the Dataverse software at the Google cloud infrastructure of CESSDA ERIC. A cloud installation makes it possible to automate the installation and keep the application up and running, for instance by scaling up or down resources when needed. Another advantage of a cloud orchestrator is the ability to start a new component or part of the application, if it should fail for some reason.

Furthermore, task 5.2 developed several additional functionalities to the Dataverse software to make the software more compliant to the needs of the SSH communities in Europe.

This document describes the results of the accomplished work, and refers to technical details published in GitHub repositories. Already many of the results of task 5.2 are used by the European and Global Dataverse community and some functionalities are integrated in the new versions of the Dataverse master branch of Harvard.

[1] Dataverse website: https://dataverse.org/about [21.02.2022]

Abbreviations and Acronyms

API	Application Programming Interface
AUSSDA	The Austrian Social Science Data Archive
AWS	Amazon Web Services
CEDAR	Center for Expanded Data Annotation and Retrieval
CESSDA	Consortium of European Social Science Data Archives
CCR	CLARIN Concept Registry
CD	Continuous Deployment
CI	Continuous Integration
CI/CD	Continuous Integration and Deployment
CMDI	Component MetaData Infrastructure
CLARIN	Common Language Resources and Technology Infrastructure
CMM	CESSDA Metadata Model
CNR	National Research Council of Italy
CV	Controlled Vocabulary
DANS	Data Archiving and Networked Services
DARIAH	Digital Research Infrastructure for the Arts and Humanities
distros	distributions
E-HRIS	European Research Infrastructure for Heritage Science
ELG	European Language Grid
EOSC	European Open Science Cloud
ERIC	European Research Infrastructure Consortium
FAIR	Findable, Accessible, Interoperable and Reusable
GCP	Google Cloud Platform
GDCC	Global Dataverse Community Consortium
GDPR	General Data Protection Regulation
GitHub	Repository for software code
IaC	Infrastructure as Code
ISTI	Institute of Information Science and Technologies "Alessandro Faedo"
IQSS	Institute for Quantitative Social Science
JSON	JavaScript Object Notation

KNAW	Royal Netherlands Academy of Arts and Sciences
K8s	Kubernetes
LRS	Language Resource Switchboard
NER	Named Entity Recognition
ORCID	Open Researcher and Contributor ID
PSNC	Poznan Supercomputing and Networking Center
Solr	Searching On Lucene w/ Replication
SSH	Social Sciences and Humanities
SSH	Secure Shell
SSL	Secure Sockets Layer
UI	User Interface
VLO	Virtual Language Observatory

Table of Contents

1. Introduction **7**

2.1 Background Information 9

2.2 Features of the “Archive in a BOX” 9

2.3 Usage of the “Archive in a BOX” 12

3. Centralised installation in the cloud **13**

3.1 Background information 13

3.2 Minimal generic K8s solution 14

3.3 CESSDA Technical Infrastructure 15

4. Additional functionality **17**

4.1 Background information 17

4.2 pyDataverse 18

4.3 Workflow to translate the Graphical User Interface to languages other than English 18

4.4 Inclusion of metadata standards that are important to the European SSH research communities 19

4.4.1 CESSDA Metadata Model (CMM) 20

4.4.2 CLARIN Component MetaData Infrastructure (CMDI) 20

4.5 Plugin for Controlled vocabulary support 22

4.6 Plugin with Taverna workflows 24

4.7 Development of previewers 24

4.8 Integration with CLARIN’s Language Resource Switchboard 26

4.9 Integration with Apache Superset 27

5. Conclusion **30**

6. References **31**

1. Introduction

Within the task 5.2 of the SSHOC project, repository software is being developed based on Dataverse[2], for the sharing and publication of research data within the Social Sciences and Humanities (SSH) domain. Dataverse is open-source research data repository software developed by the Institute for Quantitative Social Science (IQSS), Harvard University. Increasingly more developers from the Dataverse community improve the software together with Harvard.

The group of developers and research data specialists of task 5.2, worked on functionalities to make the Dataverse repository software more compliant with the needs of the European SSH community. This document describes the *'Archive in a Box' repository software and the proof of concept of an installation in the cloud*.

To make it easier for institutes with lack of technical staff to install the Dataverse repository software, task 5.2 developed an 'Archive in a Box' solution. After downloading, this package does an automatic installation and setup of the complete repository infrastructure. The "Archive in a Box" will be available via the SSH Open Marketplace[3]

Additionally, to the development of an automated installation package, task 5.2 worked on a proof of concept of a cloud installation of the Dataverse software at the Google cloud infrastructure of CESSDA. A cloud installation makes it possible to automate the installation and keep the application up and running, for instance by scaling up or down resources when needed. Another advantage of the cloud orchestrator is the ability to start a new component or part of the application, if it should fail for some reason. A cloud installation could be used by European Research Infrastructure Consortiums (ERICs) to implement a service that can be used by all their partners. Such a centralised European Open Science Cloud Service (EOSC) has economies of scale. Task 5.2 only looked at the technical aspects of such a service.

Furthermore, task 5.2 developed several additional functionalities to the Dataverse software to make it more compliant with the needs of the SSH communities in Europe. The following functionalities were developed:

- Workflow to translate the Graphical User Interface to languages other than English;
- Inclusion of metadata standards that are important to the European SSH research communities;
- pyDataverse;
- Plugin for controlled vocabulary support;

[2] https://dataverse.org/about [21.02.2022]
[3] https://marketplace.sshopencloud.eu/ [21.02.2022]

- Plugin with Taverna workflows;
- Development of previewers;
- Integration with CLARIN's Language Resource Switchboard;
- Integration with Apache Superset.

This document describes the features of the developed functionalities. Links to GitHub are available for more detailed technical specifications.

Dataverse is open-source software and all code developed in task 5.2 is available through GitHub repositories. Since SSHOC is a project having a determined timeframe of implementation, the SSHOC GitHub repository sustainability was enhanced by including the code in the repository of the Global Dataverse Community Consortium (GDCC)[4]. The GDCC coordinates worldwide contributions to the Dataverse project. Some code is also integrated in the master branch versions of Dataverse. More details of sustainability aspects of the work of task 5.2 will be described in deliverable D5.6.

[4] https://dataversecommunity.global [21.02.2022]

2. The Archive in a Box

2.1 Background Information

The “Archive in a Box”[5] is a software package intended for research organisations and universities, which intend to run their own instance of a community-based data repository to make their data FAIR (Findable, Accessible, Interoperable and Reusable). The idea behind the "Archive in a Box" is simple: it should be doing an automatic installation and setting up of the complete infrastructure without any extra efforts. This would be especially useful for institutions with limited technical resources.

The software package relies on container and container orchestration technologies like Docker[6] and Kubernetes[7], and can install and manage all dependencies without human interaction. The “Archive in a Box” uses Docker Compose[8] [9], a tool for defining and running multi-container Docker applications and configuring application services. All networking issues, such as domain name setup, Secure Sockets Layer (SSL)[10] certificates and routing, are carried out by Traefik[11], a leading reverse proxy and load balancer.

The demonstration version of Dataverse is available out of the box after completing the installation on a local computer or Virtual Machine. It will be shipped with FAKE persistent identifiers[12], language switch and various content previewers, and other components integrated in the infrastructure. This default installation could be done by people without technical background and allows extensive testing of the basic functionality without spending time on the system administration tasks related to the Dataverse setup.

2.2 Features of the “Archive in a BOX”

The “Archive in a Box” has the following features:

- Fully automatic Dataverse deployment with Traefik proxy on the selected domain name and with automatically generated SSL certificates;
- Dataverse configuration managed through environmental file .env[13]

[5] https://github.com/IQSS/dataverse-docker [21.02.2022]
[6] https://www.docker.com [21.02.2022]
[7] https://kubernetes.io [21.02.2022]
[8] https://docs.docker.com/compose/ [21.02.2022]
[9] Kubernetes is not needed for local installation, but it can be used when an organisation has access to a cloud environment.
[10] https://en.wikipedia.org/wiki/Transport_Layer_Security#SSL_1.0,_2.0,_and_3.0 [21.02.2022]
[11] https://traefik.io [21.02.2022]
[12] https://guides.dataverse.org/en/latest/installation/config.html#id171 [21.02.2022]
[13] https://docs.docker.com/compose/environment-variables/ [21.02.2022]

- Different Dataverse distributions[14] allow to provide services for different use cases and using custom components;
- External controlled vocabularies support (with demonstration of CESSDA CMM metadata fields connected to Skosmos[15] framework);
- MinIO[16] storage support with standard S3[17] interface;
- External services integration with PostgreSQL[18] triggers;
- Support of custom metadata schemes (CESSDA CMM, CLARIN CMDI, ...);
- Built-in Web interface localisation uses Dataverse language pack[19] to support multiple languages out of the box.

During the installation the "Archive in a box" is using Traefik, Docker-aware reverse proxy that configures itself automatically and dynamically, and includes its own monitoring dashboard. It has functionality to generate SSL certificates automatically using Letsencrypt[20] and taking care about all infrastructural components on the network level.

The *configuration* is centrally managed by using environmental variables in the file .env[21], so administrators have no need to modify other files in the software package. It contains all necessary settings required to deploy Dataverse, for example, to set the language or web interface, establish connection to the local database, Solr[22] search engine, mail relay or external storage. This approach allows service users to create their own distributions or Dataverse distros and integrate their specific services in the same infrastructure. For example, DataverseNO[23] did the integration of the Norwegian authentication service called Feide[24], and it is part of their own "Archive in a Box" distro.

The *startup process* of the "Archive in a Box" is simplified and uses init.d[25] folder defined in .env to arrange the order of how Dataverse configuration scripts will be running in the same way as if it is implemented in the different distributions of Linux OS[26]. It contains bash scripts making the services run sequentially and allows easy customization of Dataverse instances according to the requirements of the data provider. All necessary actions like setting up a domain name and a mail relay, activate previewers,

[14] https://en.wikipedia.org/wiki/Linux_distribution [21.02.2022]
[15] https://skosmos.org/ [21.02.2022]
[16] https://min.io [21.02.2022]
[17] https://en.wikipedia.org/wiki/Amazon_S3 [21.02.2022]
[18] https://www.postgresql.org [21.02.2022]
[19] https://github.com/GlobalDataverseCommunityConsortium/dataverse-language-packs [21.02.2022]
[20] https://letsencrypt.org [21.02.2022]
[21] https://docs.docker.com/compose/environment-variables/ [21.02.2022]
[22] https://solr.apache.org/ [21.02.2022]
[23] https://dataverse.no [21.02.2022]
[24] https://www.feide.no [21.02.2022]
[25] https://www.geeksforgeeks.org/what-is-init-d-in-linux-service-management/ [21.02.2022]
[26] https://www.suse.com/support/kb/doc/?id=000016866 [21.02.2022]

webhook installation etc. can be found in this init.d folder. Previewers are enabled by the same mechanism depending on the Dataverse distribution. After being restarted, all available datasets in Dataverse will be reindexed automatically.

The *external controlled vocabularies* plugin[27] was contributed by DANS in collaboration with the GDCC, and allows the connection of Dataverse to vocabularies hosted by Skosmos[28], ORCID[29], Wikidata[30] and other service providers. The "Archive in a Box" has a basic demonstration of this feature and encourages developers from all over the world to implement their own interfaces in order to integrate Dataverse with third-party-controlled vocabularies.

Custom metadata schemes can be easily integrated in Dataverse by using the same mechanism based on the init.d folder. A new schema should be declared in the .env file first and afterwards, a script should be added to download the schema as a .tsv file and upload it in Dataverse[31]. As a demonstration of this feature, CESSDA CMM[32] and a proof of concept of CLARIN metadata[33] compliant schemes are already integrated and available in the software package, and could be activated in the .env file and in the Dataverse web interface.

Another important feature of "Archive in a Box" is *external storage support*. It has integrated High Performance, Kubernetes Native Object Storage called MinIO[34] and delivers scalable, secure, S3[35] compatible object storage to every public cloud like Amazon AWS[36], Google Cloud Platform[37] or Microsoft Azure[38]. It means Dataverse can store data in the Cloud storage instead of local file storage, and different storages could be used for the collections (subdataverses) of different data providers created within the same Dataverse instance.

There is a separate *webhook implementation*[39] for the integration of external services based on Dataverse related actions caught by a PostgreSQL trigger, like dataset modification or publication. For example, automatic FAIR assessment could be done by sending a newly created persistent identifier to the third-

[27] https://zenodo.org/record/5845540#.YgPr5e7MKjA [21.02.2022]
[28] https://skosmos.org [21.02.2022]
[29] https://orcid.org [21.02.2022]
[30] https://www.wikidata.org [21.02.2022]
[31] https://guides.dataverse.org/en/latest/admin/metadatacustomization.html [21.02.2022]
[32] https://raw.githubusercontent.com/IQSS/dataverse-docker/master/config/schemas/CESSDA_CMM.tsv [21.02.2022]
[33] https://raw.githubusercontent.com/IQSS/dataverse-docker/master/config/schemas/cmdi-oral-history.tsv [21.02.2022]
[34] https://min.io [21.02.2022]
[35] https://en.wikipedia.org/wiki/Amazon_S3 [21.02.2022]
[36] https://aws.amazon.com/ [21.02.2022]
[37] https://cloud.google.com/ [21.02.2022]
[38] https://azure.microsoft.com/en-us/ [21.02.2022]
[39] https://en.wikipedia.org/wiki/Webhook [21.02.2022]

party service when the user publishes a new dataset. There is also the possibility to integrate Dataverse with various pipelines and workflows dedicated for some specific tasks like named entity recognition (NER) in the uploaded files. It can be useful for building General Data Protection Regulation (GDPR) related workflows to get automatic checks if there are personal data present.

The *localisation of Dataverse* is realised with community maintained Dataverse language packs, see also the paragraph about Weblate. The Graphical User Interface is switched to the selected language automatically during the start-up process.

2.3 Usage of the "Archive in a BOX"

To run their Dataverse instance as a completely operational production service, data providers should fill all settings in the configuration file containing information about their domain name, DOIs settings, the language of web interface, mail relay, external controlled vocabularies, and storage. There is also the possibility to integrate Docker based custom services and create own software packages serving the specific needs of the data provider, for example, to integrate a separate Shibboleth container for federated authentication, install a new data previewer or activate a data processing pipeline.

As of February 7th 2022, there are 32 forks and 36 stars[40] on the GitHub repository containing the source code[41] of the "Archive in a Box" as a clear indication that it's being actively used by the Dataverse community. It was installed and currently tested by various organisations such as the Laboratory of Instrumentation and Experimental Particle Physics (LIP) from Portugal, the DataverseNO consortium from Norway, University of Paris (France), ISTI-CNR (Italy) and others.

[40] Which means that 36 persons gave a recommendation to use the software and 32 persons downloaded the software from the original GitHub repository and uploaded to their own GitHub.
[41] https://github.com/IQSS/dataverse-docker [21.02.2022]

3. Centralised installation in the cloud

3.1 Background information

An installation of the Dataverse software in the cloud could be used by European Research Infrastructure Consortiums (ERICs) to implement a service that can be used by all their partners. Such a centralised European Open Science Cloud Service (EOSC) has economies of scale. Task 5.2 worked on the technical aspects of such a service for CESSDA.

The CESSDA Technical Infrastructure[42] is used to deploy a 'Proof of Concept' Dataverse instance in the cloud using application containers in a cluster. The work within the project is loosely built upon results of the DataverseEU[43] project, which has ended in 2018. The CESSDA Technical Infrastructure is a Google Cloud Platform (GCP) based project. It consists of basically two major components:

- A Kubernetes (K8s) cluster on which the Dataverse applications are deployed;
- A Continuous Integration and Deployment (CI/CD) pipeline called Jenkins[44].

Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of containerized applications. It groups (Docker) containers that make up an application (i.e., Dataverse) into logical units for easy management and discovery. K8s is also an orchestrator for Docker containers. This means that containers are monitored by the system and appropriate actions can be carried out in case of an occurring event. Actions may include bringing back a fresh container online after unexpected behaviour, like a crash or hanging in a loop. Also, scaling up resources, like adding additional containers to the cluster in case of (temporary) high traffic demand, can be managed automatically by K8s.

CESSDA maintains three Kubernetes clusters: Development-cluster, Staging-cluster and Production-cluster. These K8s clusters and their usage will be discussed later.

The Jenkins instance is used both as a Continuous Integration (CI) server as well for Continuous Deployment (CD). This means that code changes in the Git repository will be automatically compiled, run and tested (CI) and if all is well, eventually deployed on the K8s cluster (CD).

The complete cloud installation is based on 'Infrastructure as Code' (IaC). This means that all resources can be re-created from scratch by running the code again. Therefore, the application is portable and can be easily transferred to other systems too.

[42] https://docs.tech.cessda.eu/platform/index.html [21.02.2022]
[43] https://aussda.at/en/about-aussda/projects/dataverse-eu/ [21.02.2022]
[44] https://www.jenkins.io [21.02.2022]

3.2 Minimal generic K8s solution

The CESSDA Technical Infrastructure sets specific requirements for the deployments, such as the use of integration tests (Jenkins / Selenium) and the use of a K8s package manager (Helm) etc. Therefore, a minimal 'generic' K8s deployment has also been made available in the SSHOC GitHub[45] repository. These resources can be deployed on any K8s cluster and run out-of-the-box, for instance on Minikube[46] for demo and testing purposes. The organisation in question then only needs to provide the mail relay to their mail server and an HTTP(S) load balancer to handle incoming and outgoing traffic to the Dataverse application. These two components are often organisation specific, therefore they are not supplied here, apart from a "mail catcher" to enable demo functionality. Apart from these applications, (file) storage is also often organisation specific, i.e. plain disk, object storage etc. Therefore, a default persistent volume claim is provided here. It is up to the organisation how to implement or provide the persistent volume that will be used by the persistent storage claim. All required K8s components, such as services, deployments, ConfigMaps[47], configuration jobs, persistent volume claims, health checks, readiness probes, etc. have been made available as K8s resources. The custom Docker images needed for Dataverse (current version: v5.9) and Solr can be built from scratch by using the supplied Docker files in the SSHOC GitHub[48] repository, or can directly be retrieved as a binary from the SSHOC Docker Hub[49] registry. By using K8s jobs, no Secure Shell access to the containers or direct Dataverse API interaction is needed. This is taken care of by running a specific job on the K8s cluster. This improves security. For the SSHOC project K8s jobs were created that can import Dataverse Custom Metadata blocks by uploading .tsv files containing the block metadata field definitions to the Dataverse API. Another job was created to link external controlled vocabularies to a metadata-field in Dataverse.

The minimal Dataverse configuration consists of three application components, or "microservices" that interact closely with each other. These are listed below:

1. Dataverse (web) application itself, including the web interface (front-end), hosted in the Payara [50]server;
2. PostgreSQL database for storage of metadata and configuration (back-end);
3. Solr search-index to facilitate search (back-end);

All K8s resources discussed here form the base of the CESSDA Technical Infrastructure deployment, which will be discussed next.

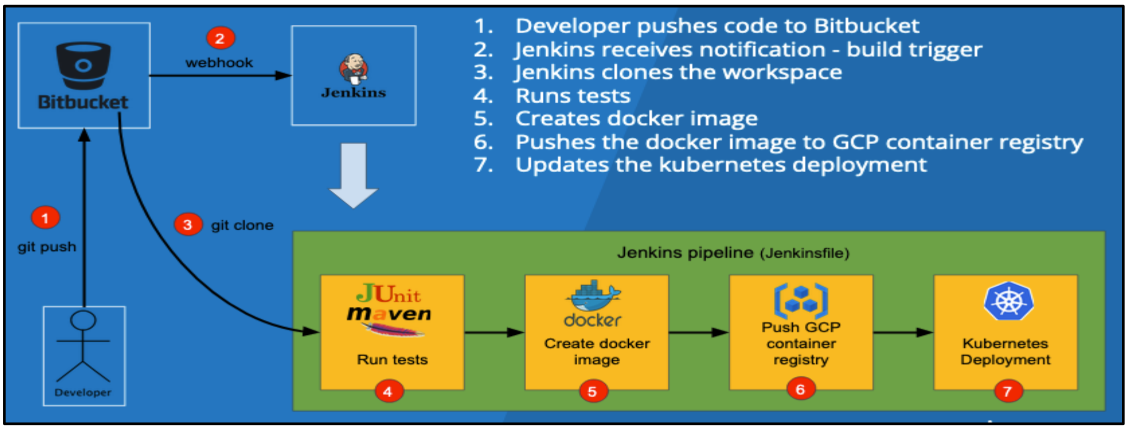
[45] https://github.com/SSHOC/dataverse-kubernetes/tree/v5.9 [21.02.2022]
[46] https://minikube.sigs.k8s.io [21.02.2022]
[47] https://kubernetes.io/docs/concepts/configuration/configmap/ [21.02.2022]
[48] https://github.com/SSHOC/dataverse-kubernetes/tree/v5.9 [21.02.2022]
[49] https://hub.docker.com/u/sshoc [21.02.2022]
[50] https://www.payara.fish [21.02.2022]

3.3 CESSDA Technical Infrastructure

On top of the described K8s base resources described above, the CESSDA Technical Infrastructure uses some custom components to meet their business policy. These include:

- Use of Jenkins CI/CD server (Jenkinsfile);
- Integration testing (Selenium automated test framework[51]);
- Use Helm[52] package manager;
- Centrally managed PostgreSQL server (hosted by Google Cloud Platform);
- Customised mail relay container, which relays all application email to their G-suite server;
- Centrally managed Load Balancer including SSL certification service. This Enables HTTPS to/from the Dataverse application.

Within this part of the SSHOC project, besides creating the K8s resources and Docker images, effort was put into defining the Jenkins pipeline and creating Selenium integration tests.



1. Developer pushes code to Bitbucket
2. Jenkins receives notification - build trigger
3. Jenkins clones the workspace
4. Runs tests
5. Creates docker image
6. Pushes the docker image to GCP container registry
7. Updates the kubernetes deployment

Figure 1: Schematic view of the complete Jenkins CI/CD pipeline used within CESSDA Infrastructure

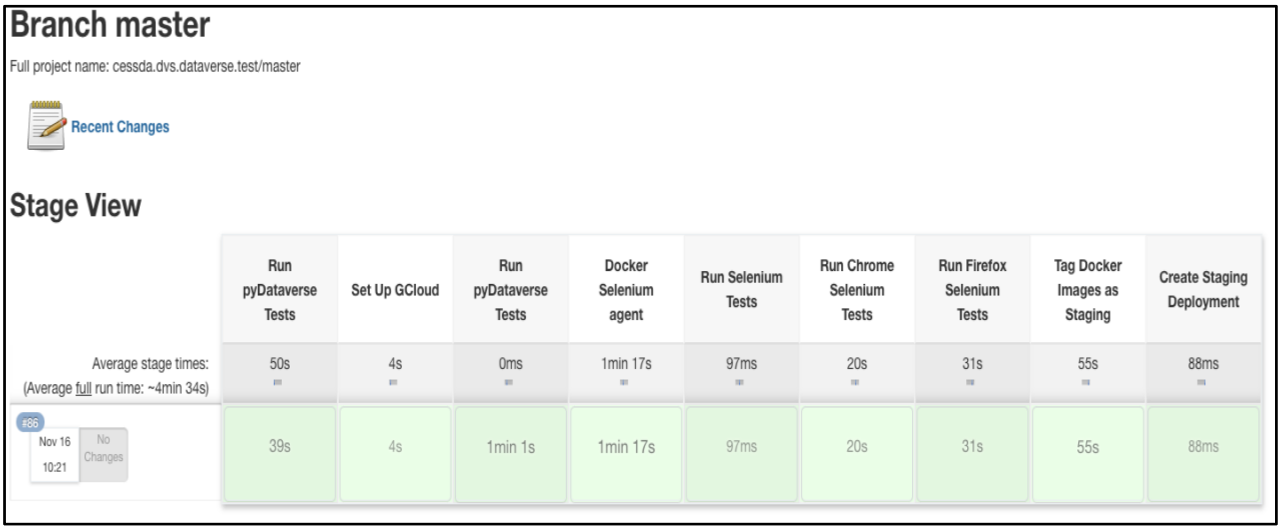
The Jenkins[53] pipeline builds, tests and creates the K8s resources when code is submitted into the Git (Bitbucket) repository. The workflow to be carried out is recorded in a so-called Jenkinsfile. This file describes the steps to be executed by Jenkins and is called a Jenkins 'pipeline'. It is available from the Git-repository for Jenkins[54]. If no errors occur during execution of the pipeline, the resources are deployed to the next K8s cluster; from 'Development' to 'Staging', thus updating the Dataverse application on Staging. The last step is to deploy the Staging application to the Production cluster. This is always a

[51] https://www.selenium.dev/ [21.02.2022]
[52] https://helm.sh/ [21.02.2022]
[53] https://jenkins.cessda.eu/ [21.02.2022]
[54] https://bitbucket.org/cessda/cessda.dvs.dataverse/src/master/Jenkinsfile [21.02.2022]

manual exercise; it is not automated. Currently this Dataverse is only available on CESSDA's Development and Staging cluster. Figure 1 shows a schematic overview of this deployment process. It is triggered each time a developer commits code changes to the git repository (Bitbucket) develop branch.

The Selenium integration tests carry out integration tests on the newly deployed Dataverse instance on the CESSDA Development-cluster. These tests use the pyDataverse[55] package which has also been developed within the SSHOC project. By using pyDataverse, known test-data is uploaded to the test Dataverse instance from within the pipeline. The created test-dataset(s) in Dataverse are then being tested by Selenium tests, also from within the pipeline.

When pipeline execution is triggered, the progress and results are reported from Jenkins. Figure 2 below shows the output of such a Jenkins job. It shows that code is being checked out from the git repository, Docker containers are created and deployed, Selenium integration tests succeed, Docker images are uploaded to the Docker registry and finally the Dataverse application is also deployed to the Staging cluster. If the pipeline fails, Docker images are not put into the registry, the application is not deployed to the Staging cluster and the Development cluster is reverted back to the previous last working version.



Branch master

Full project name: cessda.dvs.dataverse.test/master

Stage View

	Run pyDataverse Tests	Set Up GCloud	Run pyDataverse Tests	Docker Selenium agent	Run Selenium Tests	Run Chrome Selenium Tests	Run Firefox Selenium Tests	Tag Docker Images as Staging	Create Staging Deployment
Average stage times: (Average full run time: ~4min 34s)	50s	4s	0ms	1min 17s	97ms	20s	31s	55s	88ms
#86 Nov 16 10:21 No Changes	39s	4s	1min 1s	1min 17s	97ms	20s	31s	55s	88ms

Figure 2: Output of the Jenkins job in Jenkins. The defined stages in the Jenkinsfile are reported and made visible

This proof of concept of the cloud installation is also being tested on the cloud infrastructure of PSNC, the DARIAH partner within the SSHOC project.

[55] https://pydataverse.readthedocs.io/en/latest/ [21.02.2022]

4. Additional functionality

4.1 Background information

In addition to and as part of the 'Archive in a Box' and the google cloud installation, task 5.2 developed additional functionality to support the needs of the European SSH communities.

The partners of task 5.2 all belong to one of the 4 main domain-specific ERICs within SSHOC (CESSDA, CLARIN, DARIAH and E-RIHS). At the start of the project during a workshop on April 10th 2019 in The Hague, the partners discussed the preferred extra functionalities, and procedures for the development of these functionalities were established.

The list with functionalities was updated during the project on the basis of feedback of interested stakeholders. To present the work of task 5.2 and to collect feedback two webinars were organised. The first webinar[56] (March 18th, 2020) was targeted to the CESSDA community. The second webinar[57] (September 28th, 2020), was targeted to the DARIAH community. Representatives of CLARIN provided input from the CLARIN community in separate meetings.

The following functionalities were developed:

- pyDataverse;
- Workflow to translate the Graphical User Interface to languages other than English;
- Inclusion of metadata standards that are important to the European SSH research communities;
- Plugin for controlled vocabulary support;
- Plugin with Taverna workflows;
- Development of previewers;
- Integration with CLARIN's Language Resource Switchboard;
- Integration with Apache Superset.

In the following paragraphs the developed functionalities are described on a more general level. Links to GitHub are available for more technical specifications.

[56] Recordings and presentations available via: https://sshopencloud.eu/sshoc-webinar-cessda-service-providers-dataverse [21.02.2022]

[57] Recordings and slides are available via: https://sshopencloud.eu/events/sshoc-webinar-dariah-community-requirements-dataverse-repository [21.02.2022] a blog about the webinars is available at: https://www.sshopencloud.eu/news/webinar-notes-dataverse-development-sshoc [21.02.2022]

4.2 pyDataverse

pyDataverse is a Python module for Dataverse that can be used for accessing the Dataverse API's, manipulating and using the Dataverse (meta)data - dataverses, datasets and datafiles. Within task 5.2, AUSSDA extended the functionality of pyDataverse[58] to allow comprehensive testing of Dataverse instances, and created Dataverse tests[59] and Dataverse test data[60] to support this. PyDataverse had two releases (0.3.0 and 0.3.1), which added major improvements for further developments, like JSON schema validation, collecting complete Dataverse data tree, re-work of the data models, additional documentation with extensive user guides, additional API's and export/import functionality for CSV files.

Based on pyDataverse, a test repository was developed, which focus on Dataverse's most important functionality for its operation (e. g. settings, customizations, endpoints, login and other critical core functionalities) - to test Dataverse after a fresh installation, an upgrade or for frequent checks during runtime. It consists of frontend tests with Selenium and standard tests operating via different API's. It is extendable and well documented. To establish consistent development processes and standardised test procedures, a set of public test data related to Dataverse was created. It collects and shares test data for the testing of Dataverse and external tools which use Dataverse data structures and should function as a de facto standard data set for activities such as this. To support sustainability and long-term development, pyDataverse and the tests got transferred to the GDCC GitHub organisation.

4.3 Workflow to translate the Graphical User Interface to languages other than English

The Dataverse translation workflow is based on the open-source platform Weblate[61], deployed by DANS and CESSDA as a service. It's a completely free of charge web-based localization tool with tight Git integration and allows community-based translation across all languages.

CESSDA Weblate is a service used by a group of translators from the CESSDA and other SSHOC communities working in a collaborative way to get all Dataverse properties translated. It has a few key features like detection of untranslated strings and a convenient review process. Users can work simultaneously on the translation, and one reviewer can approve the changes that have been made.

[58] https://github.com/gdcc/pyDataverse [21.02.2022]
[59] https://github.com/gdcc/dataverse_tests [21.02.2022]
[60] https://github.com/AUSSDA/dataverse_testdata [21.02.2022]
[61] https://weblate.org [21.02.2022]

There are other useful features such as a glossary list, comments, and the possibility to attach relevant screenshots to make translation of every property more clear[62].

The translation of the Dataverse configuration and user interface files is rather straightforward as users can upload the current version of Dataverse files with bundle properties and Weblate can indicate which translated lines are missing and suggest translating them. Users have the possibility to translate them one by one, and to submit their translations for review when they are ready. There is also the possibility to start a translation in a new language completely from scratch, but reviewers always have to check if someone else from the community already did the translation as it can help to avoid extra work. As a part of the workflow, new translations can be uploaded in the GitHub repository automatically by Weblate or contributed manually by reviewers with the git command line. The second way is more reliable and recommended for the reviewers.

All translated files should be aggregated in the Dataverse Language Packs repository[63] where every branch corresponds to the Dataverse version. There is an automatic pipeline available in the "Archive in a Box" installation, which will download available translations and run the required set of instructions to upload the package inside of Dataverse and enable selected translations. If an administrator selected multiple languages, the language switch will be activated in the Dataverse web interface.

Weblate is a very convenient tool to keep all translations up-to-date after Dataverse releases a new version because it shows all not translated properties of the new version. There is also a "suggested" mode that allows users without appropriate permissions to change some specific translations. Reviewers have the right to accept new suggestions or decline.

In general, this workflow works well for all Dataverse properties except Solr fields. There is an extra step required to get Solr fields translated in the same way and currently it's out of the scope of this work as it's in XML format not fully supported by Weblate.
The development of this translation workflow was accompanied by the organisation of three workshops[64] for translators and a user manual.[65]

4.4 Inclusion of metadata standards that are important to the European SSH research communities

Metadata are important in order to make research data findable, interoperable and reusable; cf. the F, I and R in FAIR. The Dataverse software ships already with a set of metadata schemas which support both

[62] https://zenodo.org/record/4807371 [21.02.2022]

[63] https://github.com/GlobalDataverseCommunityConsortium/dataverse-language-packs [21.02.2022]

[64] The notes of the three events are available on the SSHOC website: https://sshopencloud.eu/news/sshoc-workshop-notes-dataverse-translation-follow-event [21.02.2022]

[65] https://zenodo.org/record/4807371#.YgQuse5KijA [21.02.2022]

generic metadata standards (e.g., Dublin Core, DataCite) and domain-specific metadata standards (e.g., ISA-Tab); cf. Dataverse User Guide[66]. The domain-specific metadata schemas are rather limited in scope, and more advanced schemas have been requested by domain-focused repositories in the Dataverse community. Fortunately, customised metadata schemas can easily be added to a Dataverse repository by installing a tsv file which defines the different fields. This feature has been exploited by task 5.2 to add support for two metadata standards used in the SSH community: 1) the CESSDA Metadata Model (CMM) and 2) the CLARIN Component MetaData Infrastructure (CMDI).

4.4.1 CESSDA Metadata Model (CMM)

CMM was developed by the Metadata Office of CESSDA with the main objective to standardise metadata for CESSDA Service Providers. CMM was built from the viewpoint of quantitative (social sciences) data and based on the DDI Lifecycle 3.2 metadata standard. Containing metadata elements, their definitions and other requirements, such as repeatability and the use of certain Controlled Vocabularies, CMM consists of 11 top elements with information about 1) Study, 2) Person(s), 3) Organisations, 4) Dataset, 5) Instrument, 6) Questions and Answers, 7) Concepts, 8) Further Documents, 9) Publications, 10) Group of Studies, and 11) Document Description ('metadata about metadata').

The standard Dataverse software contains a metadata block called 'Social Sciences metadata block', but the metadata fields in this block do not fully cover the CESSDA requirements. Therefore, a new metadata block was designed within Task 5.2. A tsv file[67] was created that can be incorporated automatically if this is indicated in the Docker configuration. Together with the use of the Controlled Vocabulary Support, this metadata block brings Dataverse closer to being compliant with the CMM.

However, there are still some technical hurdles in order to be fully compliant. The CMM requires a language tag, but it is at the moment not possible to add this to the metadata. However, within the CESSDA community there are ongoing efforts to improve the Dataverse software on this aspect. The recent contribution from the French institution Sciences Po is already in the review process and should add language attributes to the DDI metadata export to solve this issue[68].

4.4.2 CLARIN Component MetaData Infrastructure (CMDI)

CMDI was developed by CLARIN and is used as a standard for metadata provision from CLARIN centres. CMDI is not a (single) metadata standard, but rather a framework to describe and reuse metadata blueprints[69]. CMDI profiles consist of components which may be based on multiple metadata standards

[66] https://guides.dataverse.org/en/latest/user/appendix.html [21.02.2022]

[67] https://raw.githubusercontent.com/SSHOC/dataverse/3212b8c0cd44fddea5283efc9ad197e0fdd107d6/scripts/api/data/metadatablocks/CMM_Custom_MetadataBlock.tsv [21.02.2022]

[68] https://github.com/IQSS/dataverse/pull/7958 [21.02.2022]

[69] https://www.clarin.eu/content/component-metadata [21.02.2022]

(e.g., Dublin Core, DDI Codebook, ISO 639 etc.). Components are registered in the CMDI Component Registry (CCR), as illustrated in figure 3:

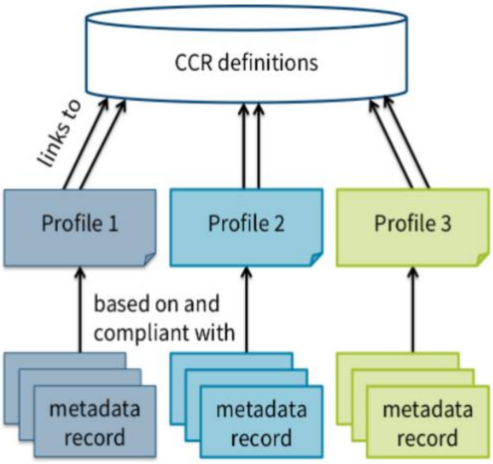


Figure 3: Component MetaData Infrastructure (CMDI)
(Image: https://www.clarin.eu/content/component-metadata)

CMDI-compatible metadata provided by CLARIN-affiliated repositories are harvested by the CLARIN Virtual Language Observatory (VLO)[70].

Two prototype sets of CMDI-compatible metadata schemas have been developed in task 5.2. The first set is based on a beta version of a set of core metadata CMDI profiles defined by the CLARIN CMDI Working Group. In addition to a general profile applying to all kinds of linguistic resources, the CLARIN Core Metadata includes profiles for the following use cases: Collections, Annotated resources, Audio-visual resources, Services & tools, and Historical documents.

The second CMDI-compliant metadata set is an implementation of the European Language Grid Metadata Schema (ELG-SHARE)[71]. The ELG is aiming at developing and deploying a scalable cloud platform to provide easy-to-integrate access to commercial and non-commercial Language Technologies for all European languages, including running tools and services as well as data sets and resources. ELG-SHARE is used for the description of all entities included in the ELG catalogue, including Language Technology Resources, both non-functional (corpora, lexica, terminologies, models, etc.) and functional (tools and cloud-based services), as well as entities related to them and involved in Language Technology at large, such as persons, organisations, projects, documents and licences. ELG-SHARE is quite complex, but provides also a minimal version comprising a set of carefully selected mandatory and recommended metadata elements. This prioritisation is reflected in the CMDI implementation in Dataverse: Mandatory ELG-SHARE elements are defined as mandatory fields in the ELG Metadata Schema in Dataverse.

[70] https://vlo.clarin.eu/;jsessionid=E5F4B24A79AE069B527936CAE07B320E?0 [21.02.2022]
[71] https://european-language-grid.readthedocs.io/en/stable/all/A2_Metadata/Metadata.html [21.02.2022]

Currently, Dataverse doesn't support an explicit distinction between recommended and optional fields. However, since metadata is filled in by the user in two rounds, a first round with essential fields, and a second round with remaining fields, the recommended ELG-SHARE fields are configured to appear in the first metadata round.

The effort to make Dataverse CMDI compliant is still a work in progress. Both sets of CMDI-compatible metadata schemas will have to be refined in collaboration with the CLARIN community and the Dataverse community, as well as aligned with the External Controlled Vocabulary support discussed in the next section. Another issue that should be addressed is improving Dataverse support for other, more complex CMDI profiles. CMDI profiles are strictly hierarchically organised based on components. In more advanced cases, this hierarchical structure can be challenging to implement into Dataverse metadata schemas using a flat tsv file. A possible way to approach this challenge is to integrate the Dataverse software with tools for metadata capture like CEDAR[72]. Finally, metadata support in Dataverse should be enhanced to make it possible to explicitly distinguish between 1) mandatory (including mandatory when applicable), 2) recommended, 3) optional fields.

In addition, Task 5.2 has developed a generic mechanism to transform existing metadata into another format by applying external XSLT mappings. It's being reused by different communities and individual developers. For example, the CLARIN community is working on the conversion of Dataverse metadata into CMDI-compatible XML metadata files which could be used within their tools. This mechanism is implemented in the same way like GitHub webhooks[73] and allows any kind of integration, not only metadata transformation.

4.5 Plugin for Controlled vocabulary support

A controlled vocabulary (CV) is a restricted list of words or terms used for labelling, indexing or categorising. The usage of controlled vocabularies, helps to standardise metadata at the moment of its creation. Dataverse already supported the use of internally managed controlled vocabularies as part of metadata blocks. Any metadata block field can be associated with a fixed list of terms that are the allowed values for that field.

However, support for external controlled vocabulary sources was one of the most requested functionalities by the SSH community to develop. Task 5.2 of the SSHOC project has developed the initial Dataverse plugin enabling external controlled vocabulary support from Skosmos[74] which is widely used

[72] https://hdl.handle.net/10037/21462 [21.02.2022]
[73] https://docs.github.com/en/developers/webhooks-and-events/webhooks/about-webhooks [21.02.2022]
[74] https://skosmos.org [21.02.2022]

within European scientific organisations to host controlled vocabularies[75]. It was extended by GDCC to support more sources like ORCID[76] and make this plugin more generic. The CV support functionality is part of the master branch of Dataverse version 5.7[77] onwards, officially released in October, 2021.

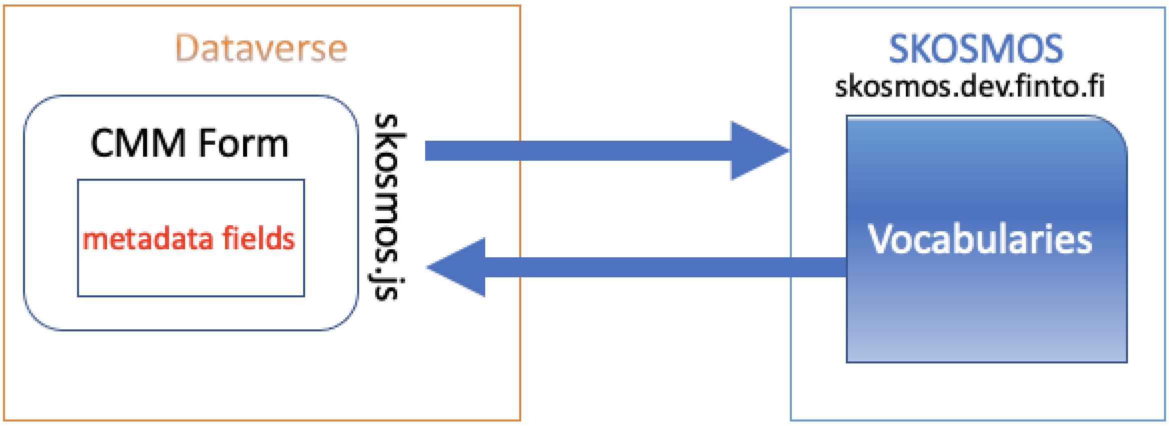


Figure 4: Schematic overview of the architecture of the controlled vocabulary support

The generic external vocabulary support mechanism makes use of service-specific scripts (Figure 4), and a custom json[78] configuration setting *which describes the* specification of how fields in Dataverse metadata blocks are to be associated with specific services and vocabularies. The design of this feature follows the client-server architecture[79] where the frontend can be hosted completely independently and maintained by GDCC[80] and doesn't require any integration with the backend service consisting of the Dataverse core. This gives the opportunity for the community to implement their own interfaces on the frontend side for the integration of new vocabularies, and customise the deposit form of the Dataverse web interface without waiting for a new Dataverse release. There are other new developments using this distributed approach, for example, searching for authors in external authority sources and fill in their names in the metadata[81].

Administrators have the ability to link specific metadata fields to externally maintained controlled vocabularies. By enabling the controlled vocabulary support in the Dataverse setting to associate specific metadata fields with third-party vocabulary services, Dataverse provides an easy way for users to select values from those vocabularies.

When a depositor completes metadata fields that use the controlled vocabulary feature, Dataverse will communicate with the third-party vocabulary service through a given protocol like Skosmos or ORCID.

[75] https://zenodo.org/record/5845540#.YgRJ8-5KijA [21.02.2022]
[76] https://orcid.org/ [21.02.2022]
[77] https://dataverse.org/blog/dataverse-software-57-release [21.02.2022]
[78] https://www.json.org/json-en.html [21.02.2022]
[79] https://en.wikipedia.org/wiki/Client%E2%80%93server_model [21.02.2022]
[80] https://github.com/gdcc/dataverse-external-vocab-support [21.02.2022]
[81] https://github.com/gdcc/dataverse-external-vocab-support/pull/9 [21.02.2022]

The communication is facilitated by a JavaScript interface (skosmos.js) and could be easily extended with support of other vocabulary services without making changes in the Dataverse core.

More information about the controlled vocabulary support is available on the SSHOC GitHub[82] and the GDCC GitHub[83]

4.6 Plugin with Taverna workflows

The Dataverse-Taverna plugin enables Taverna Workbench[84], (a domain-independent suite of tools used to design and execute data-driven workflows) to load or save a workflow description in a dataverse repository. The software is composed of two main components: API publishing functionalities as Web Services and a Taverna Workbench plugin implementing the integration layer with the SSHOC Repository.

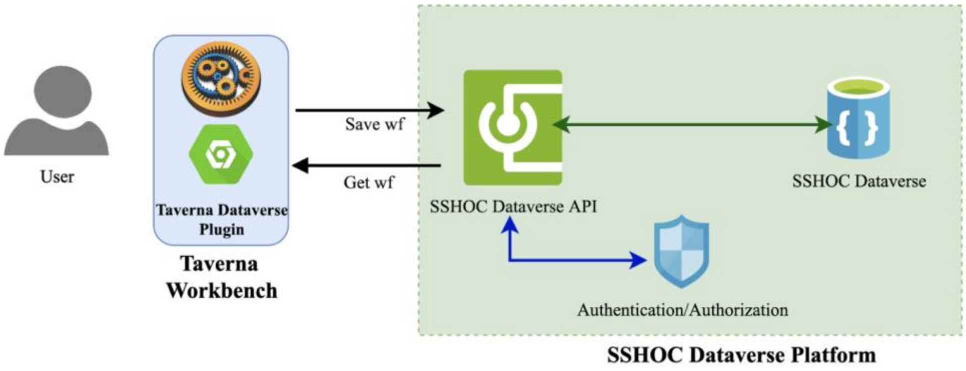


Figure 5: Architecture Dataverse-Taverna plugin

The development is in progress (alpha release), the source code is available on the SSHOC Dataverse GitHub repository[85].

4.7 Development of previewers

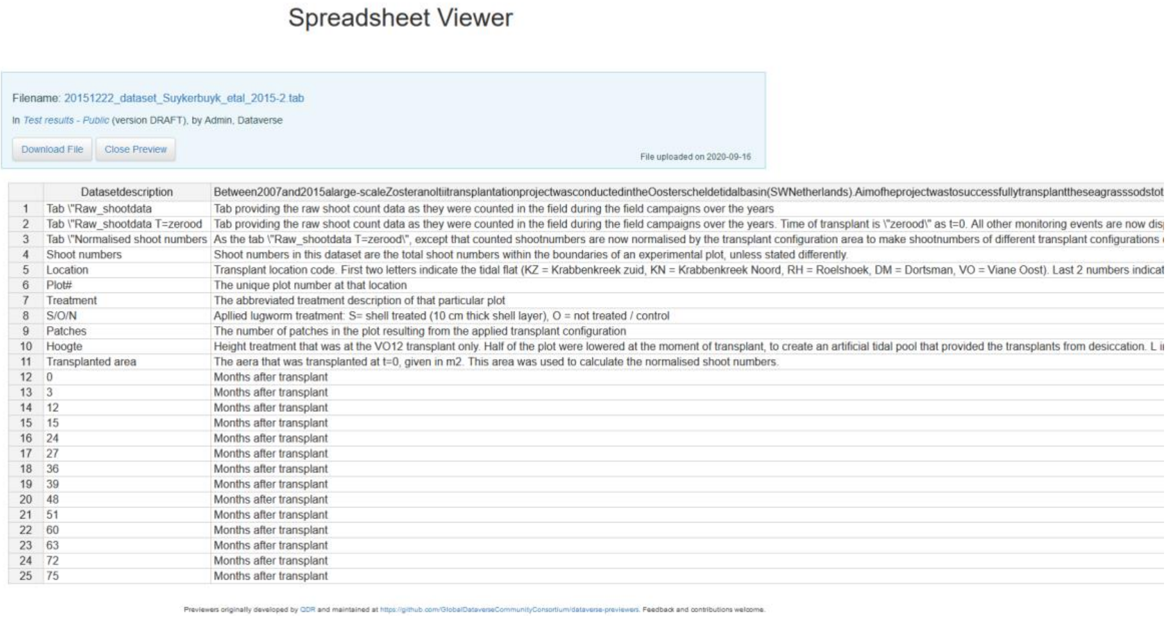
A Dataverse content previewer for spreadsheets has been selected to work on in the context of the SSHOC project, because it was frequently requested by the Dataverse community. The intention of this

[82] https://github.com/SSHOC/sshoc-dataverse-docs/tree/main/manuals/external-cvv [21.02.2022]
[83] https://github.com/gdcc/dataverse-external-vocab-support [21.02.2022]
[84] http://www.taverna.org.uk/download/workbench/ [21.02.2022]
[85] https://github.com/SSHOC/taverna-workflows [21.02.2022]

development was to help users preview data stored in their Dataverse instance, especially those stored in spreadsheet (e.g., CSV) format. The previewer has been developed by PSNC, and was proposed to be integrated with the main branch of the official Dataverse source code. The request for integration has been approved by the Dataverse community and in November 2018 it was included in version 4.18 of the Dataverse software[86].

In principle, the spreadsheet viewer allows easy previewing of the tabular data ingested in dataverse, see an example in figure 6 below.

All explanations and development issues encountered in the context of the spreadsheet viewer are available in the GitHub repository[87]. One of the issues is related to development of the feature itself, and the other one is related to additional bug-fixing. The work done in the project has been also integrated in the GDCC GitHub[88], relevant code of the spreadsheet viewer is also available at the GDCC GitHub repository[89].



Spreadsheet Viewer

Filename: 20151222_dataset_Suykerbuyk_etal_2015-2.tab

In *Test results - Public* (version DRAFT), by Admin, Dataverse

File uploaded on 2020-09-16

	Datasetdescription	Between2007and2015alarge-scaleZosteranoltiitransplantationprojectwasconductedintheOosterscheldetidalbasin(SWNetherlands).AimofheprojectwastosuccessfullytranspIanttheseagrasssodstot
1	Tab \"Raw_shootdata	Tab providing the raw shoot count data as they were counted in the field during the field campaigns over the years
2	Tab \"Raw_shootdata T=zerood	Tab providing the raw shoot count data as they were counted in the field during the field campaigns over the years. Time of transplant is \"zerood\" as t=0. All other monitoring events are now dis
3	Tab \"Normalised shoot numbers	As the tab \"Raw_shootdata T=zerood\", except that counted shootnumbers are now normalised by the transplant configuration area to make shootnumbers of different transplant configurations
4	Shoot numbers	Shoot numbers in this dataset are the total shoot numbers within the boundaries of an experimental plot, unless stated differently.
5	Location	Transplant location code. First two letters indicate the tidal flat (KZ = Krabbenkreek zuid, KN = Krabbenkreek Noord, RH = Roelshoek, DM = Dortsman, VO = Viane Oost). Last 2 numbers indicat
6	Plot#	The unique plot number at that location
7	Treatment	The abbreviated treatment description of that particular plot
8	S/O/N	Apllied lugworm treatment: S= shell treated (10 cm thick shell layer), O = not treated / control
9	Patches	The number of patches in the plot resulting from the applied transplant configuration
10	Hoogte	Height treatment that was at the VO12 transplant only. Half of the plot were lowered at the moment of transplant, to create an artificial tidal pool that provided the transplants from desiccation. L i
11	Transplanted area	The aera that was transplanted at t=0, given in m2. This area was used to calculate the normalised shoot numbers.
12	0	Months after transplant
13	3	Months after transplant
14	12	Months after transplant
15	15	Months after transplant
16	24	Months after transplant
17	27	Months after transplant
18	36	Months after transplant
19	39	Months after transplant
20	48	Months after transplant
21	51	Months after transplant
22	60	Months after transplant
23	63	Months after transplant
24	72	Months after transplant
25	75	Months after transplant

Previewers originally developed by QDR and maintained at https://github.com/GlobalDataverseCommunityConsortium/dataverse-previewers. Feedback and contributions welcome.

Figure 6: Example of previewing of tabular data

[86] https://dataverse.org/blog/2019/11/dataverse-418 [21.02.2022]

[87] https://github.com/Dans-labs/dataverse-previewers/search?q=CSVPreview [21.02.2022]

[88] https://github.com/GlobalDataverseCommunityConsortium/dataverse-previewers [21.02.2022]

[89] https://github.com/GlobalDataverseCommunityConsortium/dataverse-previewers/blob/master/previewers/SpreadsheetPreview.html [21.02.2022]

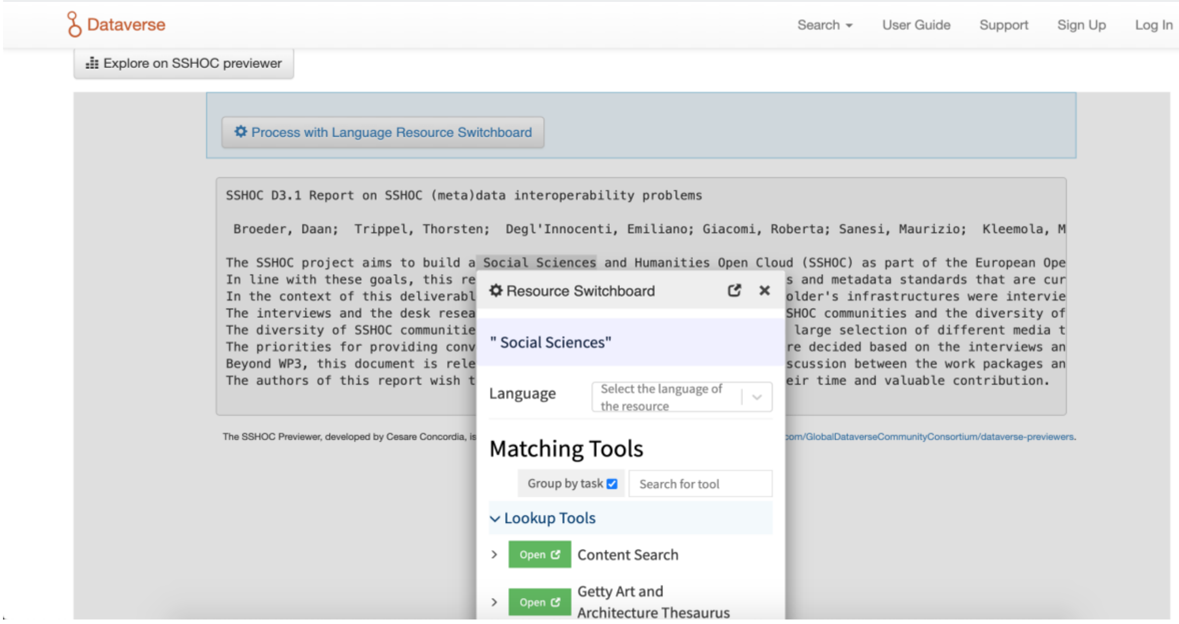
4.8 Integration with CLARIN's Language Resource Switchboard

The goal of this activity has been to integrate the CLARIN Language Resource Switchboard (LRS)[90] with the SSHOC Dataverse platform. This work has been done in collaboration with SSHOC task 3.6.

The Language Resource Switchboard (LRS) can be seen as a Virtual Tool Registry: for a given resource of a certain type, it identifies a set of tools that can process the resource, sorts the tools in terms of tasks they perform, and presents the list of tools to the user. Users can then choose and invoke a tool, and all relevant information about the resource in question is automatically passed onto the tool by the LRS. The idea behind LRS integration has been to enable a user to invoke and use the LRS functionalities from the SSHOC Dataverse platform. To enable this a plug-in was developed, called dataverse-lrs, that, using the Dataverse integration layer[91], adds to SSHOC Dataverse commands to integrate LRS functionalities. Three types of integration are provided by the lrs-plugin:

- File sent to LRS from the Dataverse Dataset View. On the right side of every resource name, there is an "Explore" icon that presents a dropdown menu with, among others, two options: "SSHOC Previewer", and "Process type file with LRS". If the second option is selected, the LRS User Interface is opened on a new web page and the resource is automatically uploaded to it for processing.
- LRS is shown in the Dataverse file preview. For two types of files, text/plain and application/pdf, a new tab called 'Preview' is added to this view. This tab shows a preview of the content of the file and contains a button that can be used to invoke the LRS; a click on this button uploads the file on the LRS and shows the LRS UI inside the Dataverse file view page.
- File previewed in a separate web page. In the Dataset view, the menu item 'SSHOC previewer' opens a new web page with a previewer that shows the content of the file. When a user selects a text fragment in the previewer, a 'pop-up menu' is displayed. If the selection consists of up to 3 words, the menu presents to the user a list of dictionaries, gazetteers, encyclopaedias and other reference tools provided by the LRS, the user can then choose and use the selected tool (see figure 7).

[90] https://switchboard.clarin.eu [21.02.2022]
[91] https://guides.dataverse.org/en/latest/admin/external-tools.html [21.02.2022]



```
SSHOC D3.1 Report on SSHOC (meta)data interoperability problems

 Broeder, Daan;  Trippel, Thorsten;  Degl'Innocenti, Emiliano; Giacomi, Roberta; Sanesi, Maurizio;  Kleemola, M

The SSHOC project aims to build a Social Sciences and Humanities Open Cloud (SSHOC) as part of the European Ope
In line with these goals, this re                    s and metadata standards that are cur
In the context of this deliverabl                    older's infrastructures were intervie
The interviews and the desk resea                    SHOC communities and the diversity of
The diversity of SSHOC communitie                     large selection of different media t
The priorities for providing conv                    re decided based on the interviews an
Beyond WP3, this document is rele                    scussion between the work packages an
The authors of this report wish t                    eir time and valuable contribution.
```

The SSHOC Previewer, developed by Cesare Concordia, is   com/GlobalDataverseCommunityConsortium/dataverse-previewers.

Figure 7: SSHOC previewer with list of matching tools

The source code and the technical documentation of the dataverse-lrs plug-in are published in the SSHOC Dataverse GitHub repository[92].

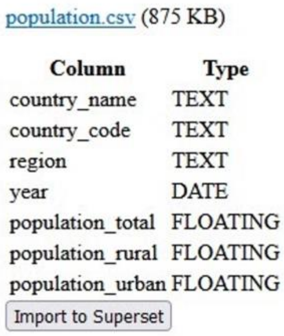
4.9 Integration with Apache Superset

Apache Superset[93] is an open-source data exploration and visualisation platform able to handle data at petabyte scale (big data). The project has seen significant contributions from leading technology companies and became a top-level project at the Apache Software Foundation in 2021[94]. The goal of the integration with Apache Superset is to help Dataverse users to create and share visualisations of their data, from simple line charts to highly detailed geospatial charts. It's integrated with Dataverse as an external tool using configuration for tabular data files (CSV, Excel) so users can choose a file in Dataverse and select the Superset from the Explore menu. They are then forwarded to the website where the Superset instance is deployed. Users get an overview of the data (see Figure 8) and have the option to import it as a dataset into Apache Superset. When the import is complete, the tool provides a link to chart creation in Apache Superset (see Figure 9), The user needs an account in Superset and a bit of knowledge about using the software. Dataverse-Superset integration can detect any charts based on the imported

[92] https://github.com/SSHOC/dataverse-lrs [21.02.2022]
[93] https://superset.apache.org/ [21.02.2022]
[94] https://en.wikipedia.org/wiki/Apache_Superset [21.02.2022]

dataset so the end-users can see the charts directly on the website, or generate an HTML snippet to embed them on their web pages like articles, personal sites, blogs, etc. (Figure 10). The code of this integration has an open-source licence and is available on the SSHOC GitHub[95].

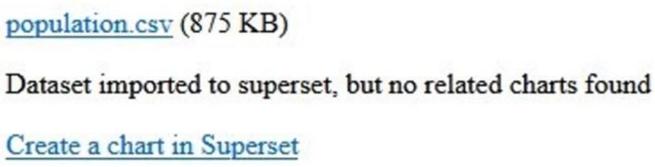
Sample screenshots of the Dataverse-Superset Integration module in figures 8 -10 below:



population.csv (875 KB)

Column	Type
country_name	TEXT
country_code	TEXT
region	TEXT
year	DATE
population_total	FLOATING
population_rural	FLOATING
population_urban	FLOATING

Figure 8: Summary displayed to the user before importing a dataset to Apache Superset module

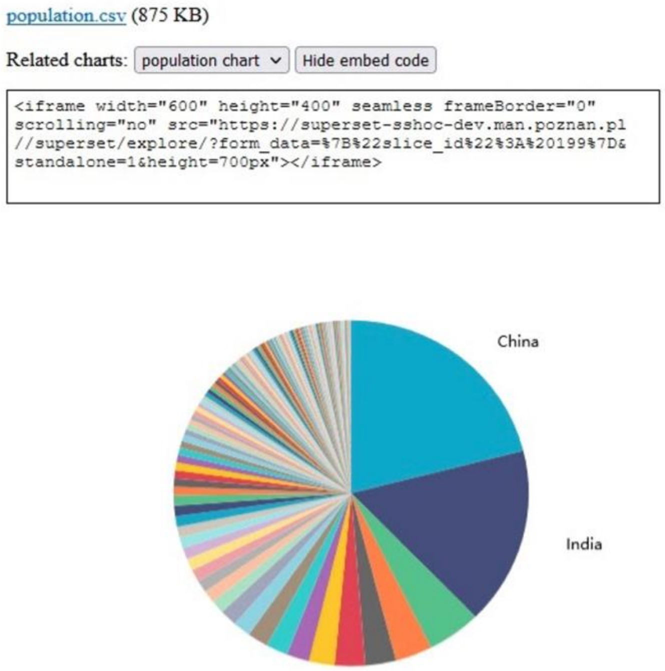


population.csv (875 KB)

Dataset imported to superset, but no related charts found

Figure 9: Chart creation link for imported dataset

[95] https://github.com/SSHOC/dataverse-superset [21.02.2022]



population.csv (875 KB)

Related charts:

```
<iframe width="600" height="400" seamless frameBorder="0"
scrolling="no" src="https://superset-sshoc-dev.man.poznan.pl
//superset/explore/?form_data=%7B%22slice_id%22%3A%20199%7D&
standalone=1&height=700px"></iframe>
```

Figure 10: Presentation of existing Superset chart and HTML snippet to embed it on external websites

5. Conclusion

SSHOC Task 5.2 contributed successfully to the community driven open-source project Dataverse. It developed an 'Archive in a Box' software solution, to make the installation of Dataverse repository software for research data easier for institutes with a lack of technical resources. Additionally, it developed a centralised cloud installation at the Google cloud infrastructure of CESSDA (as a proof of concept), to automate the installation and management. Furthermore, several additional functionalities were developed to make the Dataverse software more compliant to the needs of the SSH communities in Europe.

This document describes the work done, on a more general level and refers to technical details published in GitHub repositories. Already many results of task 5.2 are used by the Dataverse community via the GDCC GitHub and some functionalities are integrated in the new versions of the Dataverse master branch of Harvard.

6. References

- Dataverse project: https://dataverse.org/about [21.02.2022]
- SSH Open Marketplace: https://marketplace.sshopencloud.eu [21.02.2022]
- Global Dataverse Community Consortium (GDCC)l https://dataversecommunity.global [21.02.2022]
- 'Archive in a Box' software: https://github.com/IQSS/dataverse-docker [21.02.2022]
- Docker: https://www.docker.com [21.02.2022]
- Kubernetes: https://kubernetes.io [21.02.2022]
- Docker Compose: https://docs.docker.com/compose/ [21.02.2022]
- Secure Sockets Layer: https://en.wikipedia.org/wiki/Transport_Layer_Security#SSL_1.0,_2.0,_and_3.0
- [21.02.2022]
- Traefik: https://traefik.io [21.02.2022]
- Dataverse Installation guide: https://guides.dataverse.org/en/latest/installation/config.html#id171 [21.02.2022]
- Environment variables: https://docs.docker.com/compose/environment-variables/ [21.02.2022]
- Linux: https://en.wikipedia.org/wiki/Linux_distribution [21.02.2022]
- SKOSMOS: https://skosmos.org/ [21.02.2022]
- MinIO: https://min.io [21.02.2022]
- Amazon S3 storage: https://en.wikipedia.org/wiki/Amazon_S3 [21.02.2022]
- PostgreSQL: https://www.postgresql.org [21.02.2022]
- Translations User Interface Dataverse: https://github.com/GlobalDataverseCommunityConsortium/dataverse-language-packs [21.02.2022]
- Let's Encrypt: https://letsencrypt.org [21.02.2022]
- Solr: https://solr.apache.org/ [21.02.2022]
- Dataverse Norway: https://dataverse.no [21.02.2022]
- Norwegian authentication service: https://www.feide.no [21.02.2022]
- Init.d: https://www.geeksforgeeks.org/what-is-init-d-in-linux-service-management/ [21.02.2022]
- Linux OS: https://www.suse.com/support/kb/doc/?id=000016866 [21.02.2022]
- External controlled vocabularies plugin: https://zenodo.org/record/5845540#.YgPr5e7MkjA [21.02.2022]
- SKOSMOS: https://skosmos.org [21.02.2022]
- ORCID: https://orcid.org [21.02.2022]
- Wikidata: https://www.wikidata.org [21.02.2022]
- Custom Metadata Schemes: https://guides.dataverse.org/en/latest/admin/metadatacustomization.html [21.02.2022]
- CESSDA CMM tsv: https://raw.githubusercontent.com/IQSS/dataverse-docker/master/config/schemas/CESSDA_CMM.tsv [21.02.2022]
- CLARIN CMDI tsv: https://raw.githubusercontent.com/IQSS/dataverse-docker/master/config/schemas/cmdi-oral-history.tsv [21.02.2022]
- MinIO: https://min.io [21.02.2022]
- Object storage: https://en.wikipedia.org/wiki/Amazon_S3 [21.02.2022]

- Amazone AWS storage: https://aws.amazon.com/ [21.02.2022]
- Google cloud platfrom: https://cloud.google.com/ [21.02.2022]
- Microsoft Azure: https://azure.microsoft.com/en-us/ [21.02.2022]
- Webhook: https://en.wikipedia.org/wiki/Webhook [21.02.2022]
- CESSDA technical Infrastructure: https://docs.tech.cessda.eu/platform/index.html [21.02.2022]
- DataverseEU project: https://aussda.at/en/about-aussda/projects/dataverse-eu/ [21.02.2022]
- Jenkins: https://www.jenkins.io [21.02.2022]
- SSHOC GitHub repository: https://github.com/SSHOC/dataverse-kubernetes/tree/v5.9 [21.02.2022]
- Minikube: https://minikube.sigs.k8s.io [21.02.2022]
- Kubernetes: https://kubernetes.io/docs/concepts/configuration/configmap/ [21.02.2022]
- SSHOC docker hub: https://hub.docker.com/u/sshoc [21.02.2022]
- Payara: https://www.payara.fish [21.02.2022]
- Selenium: https://www.selenium.dev/ [21.02.2022]
- Helm: https://helm.sh/ [21.02.2022]
- Jenkins CESSDA: https://jenkins.cessda.eu/ [21.02.2022]
- CESSDA Bitbucket: https://bitbucket.org/cessda/cessda.dvs.dataverse/src/master/Jenkinsfile [21.02.2022]
- PyDataverse documentation: https://pydataverse.readthedocs.io/en/latest/ [21.02.2022]
- Recordings and presentations SSHOC Dataverse webinar for CESSDA: https://sshopencloud.eu/sshoc-webinar-cessda-service-providers-dataverse [21.02.2022]
- Recordings and slides SSHOC Dataverse webinar for DARIAH: https://sshopencloud.eu/events/sshoc-webinar-dariah-community-requirements-dataverse-repository [21.02.2022]
- PyDataverse: https://github.com/gdcc/pyDataverse [21.02.2022]
- Dataverse tests: https://github.com/gdcc/dataverse_tests [21.02.2022]
- Dataverse test data: https://github.com/AUSSDA/dataverse_testdata [21.02.2022]
- Weblate translation software: https://weblate.org [21.02.2022]
- User guide translation GUO: https://zenodo.org/record/4807371 [21.02.2022]
- Translation files GUI: https://github.com/GlobalDataverseCommunityConsortium/dataverse-language-packs [21.02.2022]
- Notes translation workshops: https://sshopencloud.eu/news/sshoc-workshop-notes-dataverse-translation-follow-event [21.02.2022]
- Dataverse User Guide: https://guides.dataverse.org/en/latest/user/appendix.html [21.02.2022]
- Science PO pull request: https://github.com/IQSS/dataverse/pull/7958 [21.02.2022]
- CLARIN CMDI: https://www.clarin.eu/content/component-metadata [21.02.2022]
- CLARIN Virtual Language Observatory (VLO): https://vlo.clarin.eu/;jsessionid=E5F4B24A79AE069B527936CAE07B320E?0 [21.02.2022]
- European Language Grid Metadata Schema: https://european-language-grid.readthedocs.io/en/stable/all/A2_Metadata/Metadata.html [21.02.2022]
- Metadata Schemas in Dataverse: https://munin.uit.no/handle/10037/21462 [21.02.2022]
- Webhooks: https://docs.github.com/en/developers/webhooks-and-events/webhooks/about-webhooks [21.02.2022]
- Proposal on the ontologies and external controlled vocabularies support in Dataverse https://zenodo.org/record/5845540#.YgRJ8-5KijA [21.02.2022]

- Dataverse version 5.7: https://dataverse.org/blog/dataverse-software-57-release [21.02.2022]
- JSON: https://www.json.org/json-en.html [21.02.2022]
- Client-server Model: https://en.wikipedia.org/wiki/Client%E2%80%93server_model [21.02.2022]
- GDCC GitHub external vocabularies: https://github.com/gdcc/dataverse-external-vocab-support [21.02.2022]
- SSHOC GitHub external vocabularies: https://github.com/SSHOC/sshoc-dataverse-docs/tree/main/manuals/external-cvv [21.02.2022]
- Taverna workbench: http://www.taverna.org.uk/download/workbench/ [21.02.2022]
- Taverna Plugin: https://github.com/SSHOC/taverna-workflows [21.02.2022]
- Dataverse version 4.18: https://dataverse.org/blog/2019/11/dataverse-418 [21.02.2022]
- Dataverse previewers: https://github.com/GlobalDataverseCommunityConsortium/dataverse-previewers [21.02.2022]
- CLARIN Switchboard: https://switchboard.clarin.eu [21.02.2022]
- Dataverse Guide external Tools https://guides.dataverse.org/en/latest/admin/external-tools.html [21.02.2022]
- SSHOC GitHub repository: https://github.com/SSHOC/dataverse-lrs [21.02.2022]
- Apache Superset: https://superset.apache.org/ [21.02.2022]
- SSHOC GitHub repository, Apache Superset: https://github.com/SSHOC/dataverse-superset [21.02.2022]

List of Figures

Figure 1: Schematic view of the complete Jenkins CI/CD pipeline used within CESSDA Infrastructure.

Figure 2: Output of the Jenkins job in Jenkins. The defined stages in the Jenkinsfile are reported and made visible.

Figure 3: Component MetaData Infrastructure (CMDI)

Figure 4: Schematic overview of the architecture of the controlled vocabulary support

Figure 5: Architecture Dataverse-Taverna plugin

Figure 6: Example of previewing of tabular data

Figure 7: SSHOC previewer with list of matching tools

Figure 8: Summary displayed to the user before importing a dataset to Apache Superset module.

Figure 9: Chart creation link for imported dataset

Figure 10: Presentation of existing Superset chart and HTML snippet to embed it on external websites