Publications Office
of the European Union

Initiative for digital transformation in the Metadata and Reference Data Sector of the Publications Office of the European Union

Installation guide for the RDF validation service

This report was prepared for the Publications Office of the European Union by Infeurope.

Document metadata

Reference	WP-RD-8: Installation guide for the RDF validation service
Corporate Author	Publications Office of the European Union
Author	Eugeniu Costetchi
Reviewers	Denis Dechandon and Willem Van Gemert
Contractor	Infeurope S.A.
Framework contract	10688/37490
Work package	WP-RD-8
Delivery date	10 January 2022
Suggested readers	technical staff, system administrators, enterprise architects, software developers

Abstract

This document provides technical guidance on how to install and configure the suite of micro-services and applications necessary for the asset metadata lifecycle process at the Standardisation Unit at the Publications Office of the European Union.

Contents

1 Introduction

The Standardisation Unit (SU) at the Publications Office of the European Union (OP) is engaged in a digital transformation process oriented towards semantic technologies. In [1] is described a working definition of the architectural stance and design decisions that are to be adopted for the asset publication life-cycle process. The report describes the baseline (current) solution and the (new) target solution for the asset publication workflow that is part of the life-cycle process.

The software components building up the target publication workflow solution have been packaged as into a suite of interconnected services.

This document describes the installation and configuration procedures along with stating the scope, and target audience.

2 Scope

This document aims at covering the installation and configuration instructions for the suite of the following software services:

1. RDF validator
2. Validator celery worker

3 Target audience

The target audience for this document comprises the following groups and stakeholders:

- Technical staff in charge of operating workflow components
- System administrators
- Enterprise architects and data governance specialists
- Documentalists involved in the reference data life-cycle
- Developers in charge of workflow and component implementation
- Third parties using the SU services and data

4 Technology background

Infrastructure and deployment configuration rely on services deployed on a CentOS system.

5 Requirements

There is a range of ports that must be available on the host machine as they will be bound to by different services. Although the system administrator may choose to change them by changing the values in of specific environment variables. The inventory of pre-configured ports is provided in Table 1.

Service name	HTTP port UI	HTTP port API
RDF validator	8010	4010
redis		6379

Table 1: Port usage inventory

The minimal hardware requirements are as follows

1. CPU: 3.2 Ghz quad core
2. RAM: 16GB
3. SDD system: 32GB
4. SDD data: 128GB

6 Installation

In order for the services to function properly a CentOS system with python version 3.6 and redis service should be setup and running with the appropriate ports and addresses configured in the environment variable file.

Copy the rdf validator zip on the system you intend to run it and unzip it.

Then change directory into the *project* folder. Makefile commands to start and stop services will be available.

To start the services using Makefile

```
make install-python-dependencies
make setup-rdfunit
make run-api
make run-ui
```

To stop the services using Makefile

```
make stop-gunicorn
```

To start services without Makefile commands

```
set -o allexport; source bash/.env; set +o allexport

python3 -m venv env
source env/bin/activate
pip install -r requirements/prod.txt
```

then start the services

```
set -o allexport; source bash/.env; set +o allexport

source env/bin/activate

celery -A validator.adapters.celery.celery_worker worker --loglevel
     ${RDF_VALIDATOR_LOG_LEVEL} --logfile ${
    RDF_VALIDATOR_CELERY_LOGS} --detach
gunicorn --timeout ${RDF_VALIDATOR_GUNICORN_TIMEOUT} --workers ${
    RDF_VALIDATOR_GUNICORN_API_WORKERS} --bind 0.0.0.0:${
    RDF_VALIDATOR_API_PORT} --reload validator.entrypoints.api.run:
    app --log-file ${RDF_VALIDATOR_API_LOGS} --log-level ${
    RDF_VALIDATOR_LOG_LEVEL} --daemon
gunicorn --timeout ${RDF_VALIDATOR_GUNICORN_TIMEOUT} --workers ${
    RDF_VALIDATOR_GUNICORN_UI_WORKERS} --bind 0.0.0.0:${
    RDF_VALIDATOR_UI_PORT} --reload validator.entrypoints.ui.run:app
     --log-file ${RDF_VALIDATOR_UI_LOGS} --log-level ${
    RDF_VALIDATOR_LOG_LEVEL} --daemon
```

To stop the services run

```
source env/bin/activate

celery -A rdf_differ.adapters.celery.celery_worker control shutdown
pkill -f gunicorn
```

7 Configuration

At deployment and at runtime, the service configurations are provided through OS environment variables available in the *.env* file. The role of the *.env* file is to enable the system administrators to easily change default configurations as necessary in the context of their environment.

The suite of micro-services is built, started and shut down via Makefile commands.

In order to avoid hard coding parameters, they are defined externally in the *.env*. Having them in a single file makes sense and it is more pragmatic, as you can see and manage all parameters in one place, add the file to the version control system (the contents of the file will evolve and be in sync with the actual code) and have different files for different environments.

The following sections describe the configuration options available for each of the services.

7.1 RDF validator

The RDF validator is an online platform for validating RDF data with SHACL shape definitions. It exposes an API and an UI. The RDF validator API is the core service providing the RDF validation functionality. The URL and port are described below, as well as the request timeout:

Description	Value	Associated variable
Service URL	http://localhost	RDF_VALIDATOR_API_LOCATION
Service API port	4010	RDF_VALIDATOR_API_PORT
Is in debug mode	False	RDF_VALIDATOR_DEBUG
Service UI port	8030	RDF_VALIDATOR_UI_PORT
Web server worker process timeout	1200	RDF_VALIDATOR_GUNICORN_TIMEOUT

Table 2: RDF validator configurations

7.2 Celery worker

Celery is a simple, flexible, and reliable distributed system to process vast amounts of messages, while providing operations with the tools required to maintain such a system. It's a task queue with focus on real-time processing.

In the RDF validator project it serves the purpose of enabling multiprocessing of validation report generation.

The RDF validator application uses the following Celery environment variables

Description	Value	Associated variable
Redis location	`redis://localhost`	RDF_VALIDATOR_REDIS_LOCATION
Redis port	6379	RDF_VALIDATOR_REDIS_PORT

Table 3: Celery environment configurations

More about the implementation of multiprocessing can be found in the *adapters/celery.py*. A fragment of how celery is used and the asynchronous diff creation is presented below:

```
celery_worker = Celery('rdf-validator-tasks',
          broker=config.RDF_VALIDATOR_REDIS_SERVICE,
          backend=config.RDF_VALIDATOR_REDIS_SERVICE)
celery_worker.conf.update(result_extended=True)

CELERY_VALIDATE_FILE = 'validate_file'


@celery_worker.task(name=CELERY_VALIDATE_FILE, bind=True)
def async_validate_file(self, uid: str, data_file: str,
    shacl_shapes: list, db_cleanup_location: str,
    application_profile: str = ''):
          ...
```

7.3 Configure and read logs

Every service provided by the RDF validator has it's own log history and is configurable through the aforementioned *.env* file. The current configuration accepts a relative path to where the logs to be written *logs/api.log*, for example.

API log example

```
[2021-12-01 15:54:39 +0000] [7] [INFO] Starting gunicorn 20.1.0
[2021-12-01 15:54:39 +0000] [7] [DEBUG] Arbiter booted
```

```
[2021-12-01 15:54:39 +0000] [7] [INFO] Listening at: http
   ://0.0.0.0:4030 (7)
[2021-12-01 15:54:39 +0000] [7] [INFO] Using worker: sync
[2021-12-01 15:54:39 +0000] [9] [INFO] Booting worker with pid: 9
[2021-12-01 15:54:39 +0000] [10] [INFO] Booting worker with pid: 10
[2021-12-01 15:54:39 +0000] [7] [DEBUG] 2 workers
[2021-12-01 15:55:13 +0000] [9] [DEBUG] GET /diffs
[2021-12-01 15:55:13 +0000] [9] [DEBUG] start get diffs endpoint
[2021-12-01 15:55:13 +0000] [9] [DEBUG] finish get diffs endpoint
```

UI log example

```
[2021-12-01 15:55:21 +0000] [10] [DEBUG] GET /tasks
[2021-12-01 15:55:21 +0000] [10] [DEBUG] request active tasks view
[2021-12-01 15:55:22 +0000] [10] [DEBUG] render active tasks view
```

The RDF validator application uses the following environment variables to define logs location:

Description	Value	Associated variable
API logs	logs/api.log	RDF_VALIDATOR_API_LOGS
UI logs	logs/ui.log	RDF_VALIDATOR_UI_LOGS
Celery logs	logs/celery.log	RDF_VALIDATOR_CELERY_LOGS

Table 4: RDF validator log configurations

8 API documentation

8.1 Get application profiles

Get application profiles names and their template variations.

URL	ACTION
`/aps`	`GET`

Response

200

```
[
  {
    "application_profile": "skos-core-en-only",
    "template_variations": [
      "html",
      "json"
    ]
  },
  {
    "application_profile": "skos-core-lang-fallback",
    "template_variations": [
      "html",
      "json"
    ]
  }
]
```

8.2 Get dataset deltas

List the existent dataset deltas

URL	ACTION
`/diffs`	`GET`

Response

200

```
[
  {
      "current_version_graph": "http://publications.europa.eu/
         resource/authority/data-theme/version/new",
      "dataset_description": null,
      "dataset_id": "/diff18H35CGpD",
      "dataset_uri": "http://publications.europa.eu/resource/
         authority/data-theme/",
      "dataset_versions": [
          "new1",
          "old1"
```

```
        ],
        "diff_date": null,
        "new_version_id": "new",
        "old_version_id": "old",
        "query_url": "http://fuseki:3030/diff18H35CGpD/sparql",
        "version_history_graph": "http://publications.europa.eu/
           resource/authority/data-theme/version",
        "version_named_graphs": [
            "http://publications.europa.eu/resource/authority/data-
               theme/version/new",
            "http://publications.europa.eu/resource/authority/data-
               theme/version/old"
        ]
    },
    {
        "current_version_graph": "http://publications.europa.eu/
           resource/authority/data-theme/version/new",
        "dataset_description": null,
        "dataset_id": "/diff2F4ZLMgNu",
        "dataset_uri": "http://publications.europa.eu/resource/
           authority/data-theme/",
        "dataset_versions": [
            "new1",
            "old1"
        ],
        "diff_date": null,
        "new_version_id": "new",
        "old_version_id": "old",
        "query_url": "http://fuseki:3030/diff2F4ZLMgNu/sparql",
        "version_history_graph": "http://publications.europa.eu/
           resource/authority/data-theme/version",
        "version_named_graphs": [
            "http://publications.europa.eu/resource/authority/data-
               theme/version/new",
            "http://publications.europa.eu/resource/authority/data-
               theme/version/old"
        ]
    }
]
```

8.3 Get dataset delta

Get specific dataset delta

URL	ACTION
`/diffs/{dataset_id}`	`GET`

Parameters

Name	Description
`dataset_id`	dataset unique name

Response

200

```
{
  "current_version_graph": "http://publications.europa.eu/resource/
     authority/data-theme/version/new",
  "dataset_description": null,
  "dataset_id": "/diff18H35CGpD",
  "dataset_uri": "http://publications.europa.eu/resource/authority/
     data-theme/",
  "dataset_versions": [
    "new1",
    "old1"
  ],
  "diff_date": null,
  "new_version_id": "new",
  "old_version_id": "old",
  "query_url": "http://fuseki:3030/diff18H35CGpD/sparql",
  "version_history_graph": "http://publications.europa.eu/resource/
     authority/data-theme/version",
  "version_named_graphs": [
    "http://publications.europa.eu/resource/authority/data-theme/
       version/new",
    "http://publications.europa.eu/resource/authority/data-theme/
       version/old"
  ]
}
```

404

```
{
  "detail": "<datasetname> does not exist.",
  "status": 404,
  "title": "Not Found",
  "type": "about:blank"
}
```

8.4 Delete dataset delta

Delete specific dataset delta

URL	ACTION
`/diffs/{dataset_id}`	`DELETE`

Parameters

Name	Description
`dataset_id`	dataset unique name

Response

200

```
"<datasetname> deleted successfully."
```

404

```
{
  "detail": "<datasetname> does not exist.",
  "status": 404,
  "title": "Not Found",
  "type": "about:blank"
}
```

8.5 Create dataset delta

Create a diff delta

URL	ACTION
`/diffs`	`POST`

Body

multipart/form-data

Name	Required	Type	Description
`dataset_id`	true	`string`	The dataset identifier. This should be short alphanumeric string uniquely identifying the dataset.
`dataset_description`	true	`string`	The dataset description. This is a free text description fo the dataset.
`dataset_uri`	true	`string`	The dataset URI. For SKOS datasets this is usually the ConceptSchema URI.
`old_version_id`	true	`string`	Identifier for the older version of the dataset.
`new_version_id`	true	`string`	Identifier for the newer version of the dataset.
`old_version_file_content`	true	`file`	The content of the old version file.
`new_version_file_content`	true	`file`	The content of the new version file.
`new_version_id`	true	`string`	Identifier for the newer version of the dataset.

Response

200

```
{
  "dataset_name": "diff2F4ZLMgNu",
  "task_id": "cee03499-41b2-41e4-ae75-95b9383eea0c"
}
```

8.6 Create a report

URL	ACTION
`/diffs/reports`	`POST`

Body

application/json

Name	Required	Type	Description
`dataset_id`	true	`string`	The dataset identifier. This should be short alphanumeric string uniquely identifying the dataset.
`application_profile`	true	`string`	The application profile identifier
`template_type`	true	`string`	The template type identifier
`rebuild`	false	`string`	Flag to signal rebuilding the report even if already exists. ("true" or "false")

Response

200

```
{
  "application_profile": "skos-core-en-only",
  "task_id": "3cf43787-18e8-4927-aa5f-198da6b8bba2"
}
```

8.7 Get report

URL	ACTION
`/diffs/report`	`GET`

Parameters

Name	Description
`dataset_id`	dataset unique name
`application_profile`	The application profile identifier
`template_type`	The template type identifier

Response

200

Report file in specified format

8.8 List active tasks

URL	ACTION
`/tasks/active`	`GET`

Response

200

```
[
  {
    "acknowledged": true,
    "args": [
        "diff2F4ZLMgNu",
        "skos-core-en-only",
        "html",
        "/usr/src/app/reports",
        "/usr/src/app/resources/templates/skos-core-en-only/
            template_variants/html",
        {
            ...
        },
        {
            "available_reports": [
                {
                    "application_profile": "diff_report",
                    "template_variations": [
                        "html"
                    ]
                }
            ],
            "current_version_graph": "http://publications.europa.eu
                /resource/authority/data-theme/version/new1",
            "dataset_description": null,
            "dataset_id": "diff2F4ZLMgNu",
            "dataset_uri": "http://publications.europa.eu/resource/
                authority/data-theme/",
            "dataset_versions": [
                "new1",
                "old1"
            ],
            "diff_date": null,
            "new_version_id": "old1",
            "old_version_id": "new1",
            "query_url": "http://fuseki:3030/diff2F4ZLMgNu/sparql",
            "version_history_graph": "http://publications.europa.eu
                /resource/authority/data-theme/version",
            "version_named_graphs": [
                "http://publications.europa.eu/resource/authority/
                    data-theme/version/new1",
                "http://publications.europa.eu/resource/authority/
                    data-theme/version/old1"
            ]
        }
    ],
    "delivery_info": {
        "exchange": "",
        "priority": 0,
        "redelivered": null,
        "routing_key": "celery"
    },
    "hostname": "celery@e5d79cc45ba2",
    "id": "3cf43787-18e8-4927-aa5f-198da6b8bba2",
    "kwargs": {},
    "name": "generate_report",
    "time_start": 1638810064.448214,
    "type": "generate_report",
    "worker_pid": 25
  }
]
```

8.9 Get task status

Get specific task status

URL	ACTION
`/tasks/{task_id}`	`GET`

Parameters

Name	Description
`task_id`	task unique id

Response

200

```
{
  "task_id": "6ce77efc-667e-4cf4-bcec-cc7870fcc2db",
  "task_result": true,
  "task_status": "SUCCESS"
}
```

8.10 Stop task execution

URL	ACTION
`/tasks/{task_id}`	`DELETE`

Parameters

Name	Description
`task_id`	task unique id

Response

200

```
{
  "message": "task 3cf43787-18e8-4927-aa5f-198da6b8bba2 set for
     revoking."
}
```

406

```
{
  "detail": "task already finished executing or does not exist",
  "status": 406,
  "title": "Not Acceptable",
  "type": "about:blank"
}
```

References

[1] E. Costetchi. Asset publication lifecycle architecture. Recommendation, Publications Office of the European Union, September 2020.