CKAN Docs

Table of contents:

CKAN Deployment roadmap

Information about extensions installed in the `main` image. More info described in the Extending the base images

> **NOTE**
>
> Switch branches to see the `roadmap` for other projects: ckan-docker/branches

Element	Description	version	Status	Remarks
Core	CKAN	2.9.9	Completed	Stable installation for version 2.9.9 (Production & Dev images) via Docker Compose based on official images). Initial configuration, basic customisation and operation guide.
Core +	Datastore	2.9.9	Completed	Stable installation (Production & Dev images) via Docker Compose.
Core +	~~Datapusher~~	0.0.19	Deprecated	Updated to xloader, an express Loader - quickly load data into DataStore.
Extension	ckanext-xloader	1.0.1	Completed	Stable installation, a replacement for DataPusher because it offers ten times the speed and more robustness
Extension	ckanext-harvest	1.5.1	Completed	Stable installation, necessary for the

Element	Description	version	Status	Remarks
				implementation of the Collector (ogc_ckan)
Extension	ckanext-geoview	0.0.20	Completed	Stable installation.
Extension	ckanext-spatial	2.0.0	Completed	Stable installation, necessary for the implementation of the Collector (ogc_ckan)
Extension	ckanext-dcat	1.1.0	Completed	Stable installation, include DCAT-AP 2.1 profile compatible with GeoDCAT-AP.
Extension	ckanext-scheming	3.0.0	WIP	Stable installation. Customised ckanext schema[^5] based on the Spanish Metadata Core with the aim of completing the minimum metadata elements included in the current datasets in accordance with GeoDCAT-AP and INSPIRE.
Extension	ckanext-resourcedictionary	v1.0.1	Completed	Stable installation. This extension extends the default CKAN Data Dictionary functionality by adding possibility to create data dictionary before actual data is uploaded to datastore.

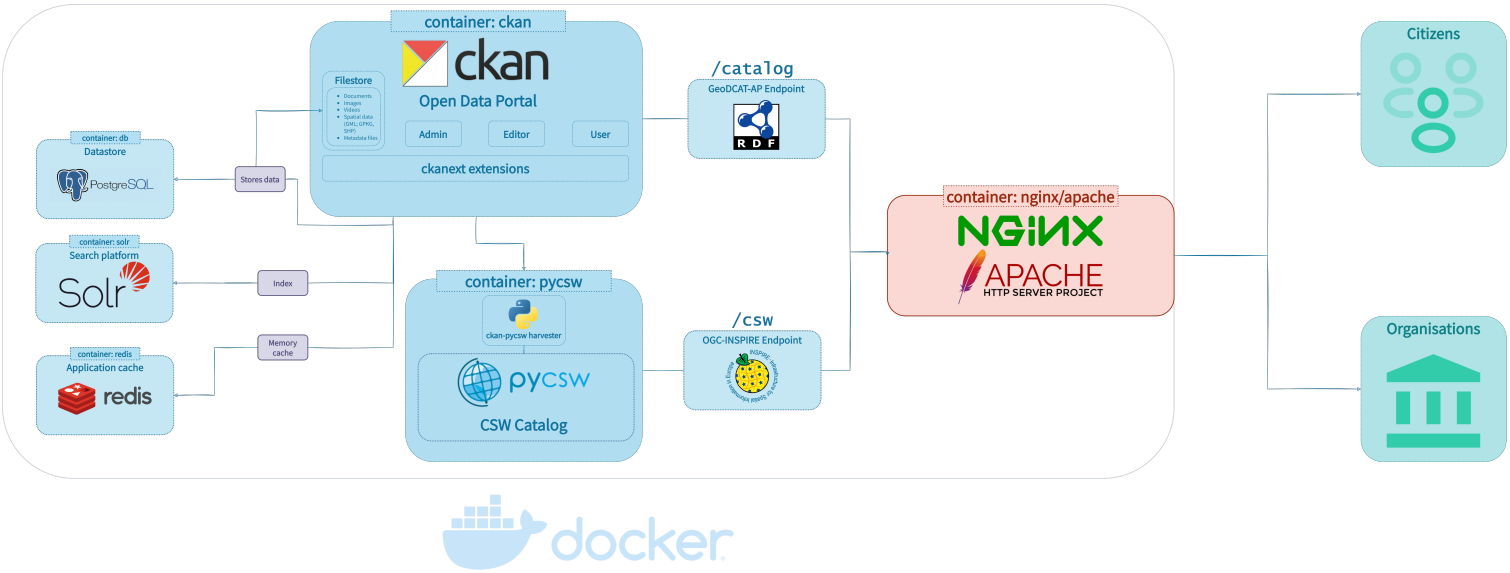
Element	Description	version	Status	Remarks
Extension	ckanext-pages	0.5.2	Completed	Stable installation. This extension gives you an easy way to add simple pages to CKAN.
Extension	ckanext-pdfview	0.0.8	Completed	Stable installation. This extension provides a view plugin for PDF files using an html object tag.
Extension	ckanext-schemingdcat	2.0.0	Completed	Stable installation for version 1.2.0, provides functions and templates specifically designed to extend ckanext-scheming and includes DCAT enhancements to adapt CKAN schema to GeoDCAT-AP and several improvements such as multilang for datasets, orgs and groups or new theming.
Extension	ckanext-fluent	1.0.1	Completed	Multilingual fields for CKAN, stable version.
Software	ckan-pycsw	main	Completed	Stable installation. PyCSW Endpoint of Open Data Portal with docker compose config. Harvest the CKAN catalogue in a CSW endpoint based on existing spatial datasets in the open data portal.

Docker and Docker Compose installation

Overview

Contains Docker images for the different components of CKAN Cloud and a Docker compose environment (based on ckan) for development and testing Open Data portals.

> **WARNING**
>
> This is a **custom installation of Docker Compose** with specific extensions for spatial data and GeoDCAT-AP/INSPIRE metadata profiles. For official installations, please have a look: CKAN documentation: Installation.



Available components:

- CKAN custom multi-stage build with spatial capabilities from ckan-docker-spatial[^1], an image used as a base and built from the official CKAN repo. The following versions of CKAN are available:

CKAN Version	Type	Docker tag	Notes
2.9.8	custom image	`ghcr.io/mjanez/ckan-spatial:ckan-2.9.8`	Stable version with CKAN 2.9.8
2.9.9	custom image	`ghcr.io/mjanez/ckan-docker:ckan-2.9.9`	Stable version with CKAN 2.9.9
2.9.9	latest custom image	`ghcr.io/mjanez/ckan-docker:master`	Latest `ckan-docker` image.

The non-CKAN images are as follows:

- PostgreSQL: Custom image based on official PostgreSQL image. Database files are stored in a named volume.
- Solr: Custom image based on official CKAN pre-configured Solr image. The index data is stored in a named volume and has a custom spatial schema upgrades. [^2]
- Redis: Standard Redis image
- NGINX: Latest stable nginx image that includes SSL and Non-SSL endpoints.
- ckan-pycsw: Custom image based on pycsw CKAN harvester ISO19139 for INSPIRE Metadata CSW Endpoint.

Optional HTTP Endpoint (`docker-compose.apache.yml`):

- `docker-compose.apache.yml`:
  - Apache HTTP Server: Custom image based on official latest stable httpd image. Configured to serve multiple routes for the ckan-pycsw CSW endpoint (`{CKAN_SITE_URL}/csw`) and CKAN (`{CKAN_SITE_URL}/catalog`). Only HTTP.

Compose files	Repository	Type	Docker tag	Size	I
`docker-compose.yml` / `docker-compose.apache.yml`	CKAN 2.9.9	custom image	`mjanez/ckan-docker:ckan-2.9.9`	800 MB	Custom I `ckan/Doc`

Compose files	Repository	Type	Docker tag	Size	I
`docker-compose.yml` / `docker-compose.apache.yml`	PostgreSQL 15.2	base image	`postgres/postgres:15-alpine`	89.74 MB	Custom I `postgres`
`docker-compose.yml` / `docker-compose.apache.yml`	Solr 8.11.1	custom image	`ckan/ckan-solr:2.9-solr9-spatial`	331.1 MB	CKAN's p spatial S
`docker-compose.yml` / `docker-compose.apache.yml`	Redis 7.0.10	base image	`redis/redis:7-alpine`	11.82 MB	-
`docker-compose.yml`	Apache HTTP Server 2.4	custom image	`httpd/httpd:2.4`	54.47 MB	Custom I `apache/D`
`docker-compose.yml`	pycsw CKAN harvester ISO19139	custom image	`mjanez/ckan-pycsw:latest`	175 MB	Custom I `ckan-pyc`
`docker-compose.apache.yml`	NGINX 1.22.1	base image	`nginx:stable-alpine`	9.74 MB	No routir Custom I `nginx/Do`

The site is configured using environment variables that you can set in the `.env` file for an NGINX and ckan-pycsw deployment (default `.env.example`), or replace it with the `.env.apache.example` for a Apache HTTP Server deployment using the Docker Compose file: `docker-compose.apache.yml`.

Requeriments and dependencies

Docusaurus consists of a set of npm packages.

> **TIP**
>
> Use the **Fast Track** to understand Docusaurus in **5 minutes** ⏱️!
>
> Use **docusaurus.new** to test Docusaurus immediately in your browser!

Requirements

- Node.js version 18.0 or above (which can be checked by running `node -v`). You can use nvm for managing multiple Node versions on a single machine installed.
  - When installing Node.js, you are recommended to check all checkboxes related to dependencies.

Scaffold project website

The easiest way to install Docusaurus is to use the command line tool that helps you scaffold a skeleton Docusaurus website. You can run this command anywhere in a new empty repository or within an existing repository, it will create a new directory containing the scaffolded files.

```
npx create-docusaurus@latest my-website classic
```

We recommend the `classic` template so that you can get started quickly, and it contains features found in Docusaurus 1. The `classic` template contains `@docusaurus/preset-classic` which includes standard documentation, a blog, custom pages, and a CSS framework (with dark mode support). You can get up and running extremely quickly with the classic template and customize things later on when you have gained more familiarity with Docusaurus.

You can also use the template's TypeScript variant by passing the `--typescript` flag. See TypeScript support for more information.

```
npx create-docusaurus@latest my-website classic --typescript
```

META-ONLY

If you are setting up a new Docusaurus website for a Meta open source project, run this command inside an internal repository, which comes with some useful Meta-specific defaults:

```
scarf static-docs-bootstrap
```

Alternative installation commands

You can also initialize a new project using your preferred project manager:

```
npm init docusaurus
```

```
my-website
├── blog
│   ├── 2019-05-28-hola.md
│   ├── 2019-05-29-hello-world.md
│   └── 2020-05-30-welcome.md
├── docs
│   ├── doc1.md
│   ├── doc2.md
│   ├── doc3.md
│   └── mdx.md
├── src
│   ├── css
│   │   └── custom.css
│   └── pages
│       ├── styles.module.css
│       └── index.js
├── static
```

```
│   └── img
├── docusaurus.config.js
├── package.json
├── README.md
├── sidebars.js
└── yarn.lock
```

CKAN: Installation and configuration

CKAN: Extensions

RHEL installation

Overview

CKAN is a powerful open data platform that provides a suite of tools to facilitate the publishing, sharing, finding and using of data. This document provides a step-by-step guide to install CKAN on a machine with Red Hat Enterprise Linux (RHEL) 9.3.

> **WARNING**
>
> This is a **custom installation** with specific extensions for spatial data and metadata GeoDCAT-AP/INSPIRE profiles. For official installations see CKAN Documentation: Installation.

Available components:

CKAN is a world-leading open source data portal platform that provides a suite of tools to streamline the publishing, sharing, finding, and using of data. CKAN is a complete out-of-the-box software solution that makes data accessible and usable – by providing tools to streamline publishing, sharing, finding and using data (including storage of data and provision of robust data APIs). CKAN is aimed at data publishers (national and regional governments, companies and organizations) wanting to make their data open and available.

The non-CKAN componentes are as follows:

- PostgreSQL: A powerful, open source object-relational database system. CKAN uses PostgreSQL to store its data. In a CKAN installation, you would need to set up a PostgreSQL database and configure CKAN to use it.
- Solr: A popular, open source search platform from the Apache Lucene project. CKAN uses Solr as its search engine. When you create or update datasets, resources, or other objects in CKAN, it updates the Solr index. Then, when you search in CKAN, it queries Solr and shows the results. The Solr index data is stored in a named volume, similar to the PostgreSQL data. In this deployment Solr has a custom spatial schema that allows for geographic searches.

- Redis: An open source, in-memory data structure store, used as a database, cache, and message broker. CKAN uses Redis as a message broker for its background jobs. When CKAN needs to perform a task that may take a long time, such as updating the search index for a large number of datasets, it adds a job to the Redis queue, which can then be processed in the background.
- NGINX: A free, open-source, high-performance HTTP server and reverse proxy. CKAN uses NGINX as a reverse proxy to route incoming HTTP requests to the CKAN application. NGINX can also serve static files, handle SSL encryption, and load balance requests between multiple CKAN instances. The NGINX configuration for CKAN includes both SSL and non-SSL endpoints, allowing for secure communication over HTTPS.

Requeriments and dependencies

Prerequisites:

Before installing CKAN, ensure you have at least the following prerequisites in place:

- RHEL 9.3 installed and configured.
- Python 3.9 or later installed.

System Preparation:

Before installing CKAN, we need to prepare the system by installing some necessary packages. The following commands should be run as root.

First, we update the system packages:

```
dnf update -y
```

Then, we install some basic packages that are necessary for the installation and operation of CKAN:

```
dnf -y install sudo nano git wget gcc openssl-devel libffi-devel make automake
cmake
```

These packages include tools for compiling code (like `gcc`), libraries for security (like `openssl-devel` and `libffi-devel`), and general utility tools (like `sudo`, `nano`, `git`, and `wget`).

Python Dependencies Installation:

CKAN is a Python application, so we need to install Python and some Python libraries. We run the following commands to install Python, pip (a Python package manager), and `libpq` (a library needed to interact with PostgreSQL, the database that CKAN uses):

```
dnf -y install python3 python3-pip libpq
```

Finally, we install `virtualenv` (a tool for creating Python virtual environments) and Ansible (a tool for system configuration automation) with pip:

```
pip3 install virtualenv
pip3 install ansible
```

With this, we have prepared the system for the installation of CKAN. In the following steps, we will proceed to install CKAN itself.

CKAN: Installation and configuration

CKAN: Extensions

api