CKAN Docs

Table of contents:

- CKAN Deployment roadmap
  - ckan-docker roadmap
- Docker and Docker Compose installation
  - Overview
  - Available components:
- Requeriments and dependencies
  - Requirements
  - Scaffold project website
- CKAN: Installation and configuration
- CKAN: Extensions
- intro
- Installation
  - Requirements
  - Scaffold project website
- ckan
- plugins
- api

CKAN Deployment roadmap

ckan-docker 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	DEV[^3]	PRO[^4]	Re
Core	CKAN	2.9.9	Completed	✔	✔	Stable installa versio (Produ Dev in via Do Compo based officia image config basic custor and op guide.
Core +	Datastore	2.9.9	Completed	✔	✔	Stable installa (Produ Dev in via Do Compo

Element	Description	version	Status	DEV[^3]	PRO[^4]	Re
Core +	~~Datapusher~~	0.0.19	Deprecated	❌	❌	Updat xloade expres - quick data ir DataS
Extension	ckanext-xloader	1.0.1	Completed	✔	✔	Stable install replac for Da becau offers the sp more robust
Extension	ckanext-harvest	1.5.1	Completed	✔	✔	Stable install neces the impler of the (ogc_c
Extension	ckanext-geoview	0.0.20	Completed	✔	✔	Stable install
Extension	ckanext-spatial	2.0.0	Completed	✔	✔	Stable install neces the impler

Element	Description	version	Status	DEV[^3]	PRO[^4]	Re
						of the (ogc_c
Extension	ckanext-dcat	1.1.0	Completed	✔	✔	Stable installa includ AP 2.1 compa with G AP.
Extension	ckanext-scheming	3.0.0	WIP	✔	✔	Stable installa Custor ckanex schem based Spanis Metad with th compl minim metad eleme includ curren datase accord with G AP and INSPIR
Extension	ckanext-resourcedictionary	v1.0.1	Completed	✔	✔	Stable installa This ex

Element	Description	version	Status	DEV[^3]	PRO[^4]	Re
						extend defaul Data D functi adding possib create dictior before data is upload datast
Extension	ckanext-pages	0.5.2	Completed	✔	✔	Stable install This e gives easy w add si pages
Extension	ckanext-pdfview	0.0.8	Completed	✔	✔	Stable install This e provid plugin files u html o tag.
Extension	ckanext-scheming_dcat	2.0.0	Completed	✔	✔	Stable install versio provid functi

Element	Description	version	Status	DEV[^3]	PRO[^4]	Re
						templa specifi design extend ckanex schem includ enhan to ada schem GeoDC and se improv such a multila datase and gr new th
Extension	ckanext-fluent	1.0.1	Completed	✔	✔	Multili fields stable
Software	ckan-pycsw	main	Completed	✔	✔	Stable install PyCSW Endpo Open Portal docker compo config the CK catalo CSW e

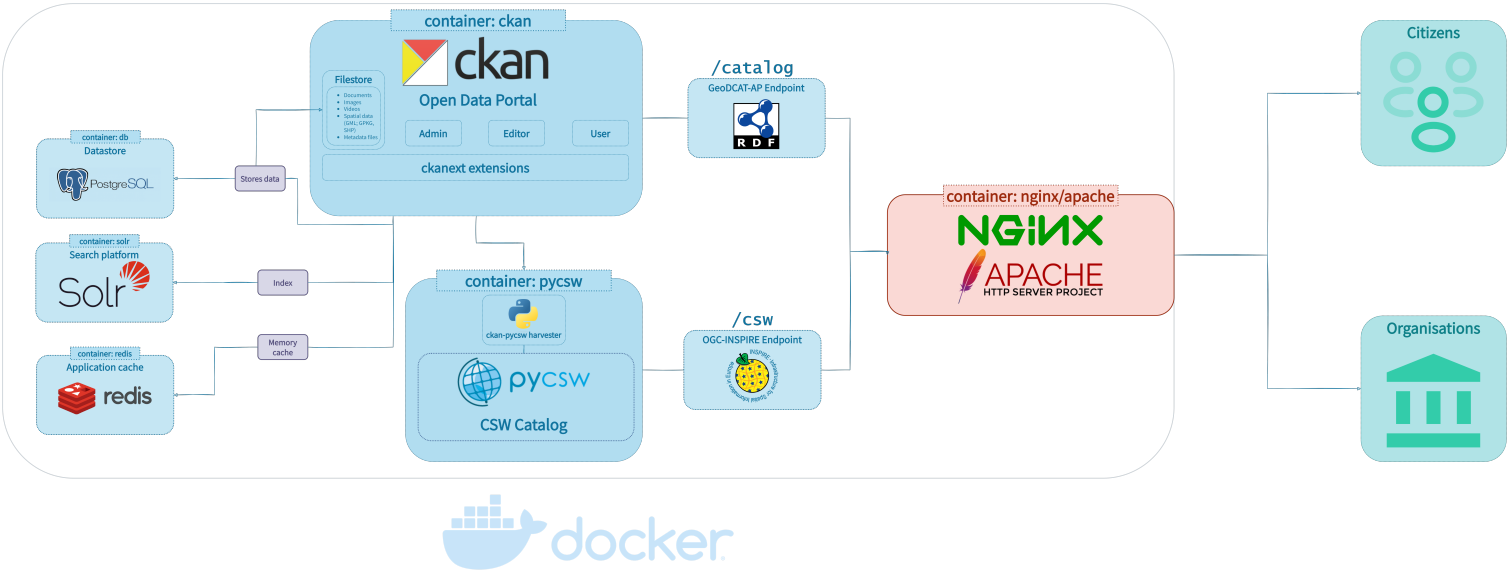
Element	Description	version	Status	DEV[^3]	PRO[^4]	Re
						based existir datase open d portal.

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

intro

Installation

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
```

<details>
<summary>Alternative installation commands</summary>

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

```
npm init docusaurus
```

</details>

```
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

plugins

api