



- [Introduction](#)
- [Overview](#)
 - [Components](#)
- [Infrastructure](#)
 - [Components](#)
 - [Deployment](#)
- [Login and Sign up](#)
 - [app.lintol.io](#)
- [Configuration and Administration](#)
 - [Licensing Lintol](#)
- [Data Profiles](#)
 - [Managing Data Profiles](#)
- [Data Processors](#)
 - [Managing Data Processors](#)
- [Add Lintol Extension to CKAN](#)
- [Contributing](#)
- [Contributor Code of Conduct](#)
 - [Our Pledge](#)
 - [Our Standards](#)
 - [Our Responsibilities](#)
 - [Scope](#)
 - [Enforcement](#)
 - [Attribution](#)
- [Sample markdown](#)
 - [Subheader 1](#)

Introduction

Welcome to the Lintol Documentation! You can use our documentation to reference the functionality that Lintol provides.

Lintol is a collection of components that work together to provide data validation and analytics services for open data. It was designed to work in conjunction with the CKAN platform.

Datasets that are uploaded to a CKAN instance can use the Lintol plugin to automatically run various validation processes on the data. This will improve the quality, speed and cost-effectiveness of data publishing. Lintol components are web-based and are configured and managed through websites within the Lintol framework.

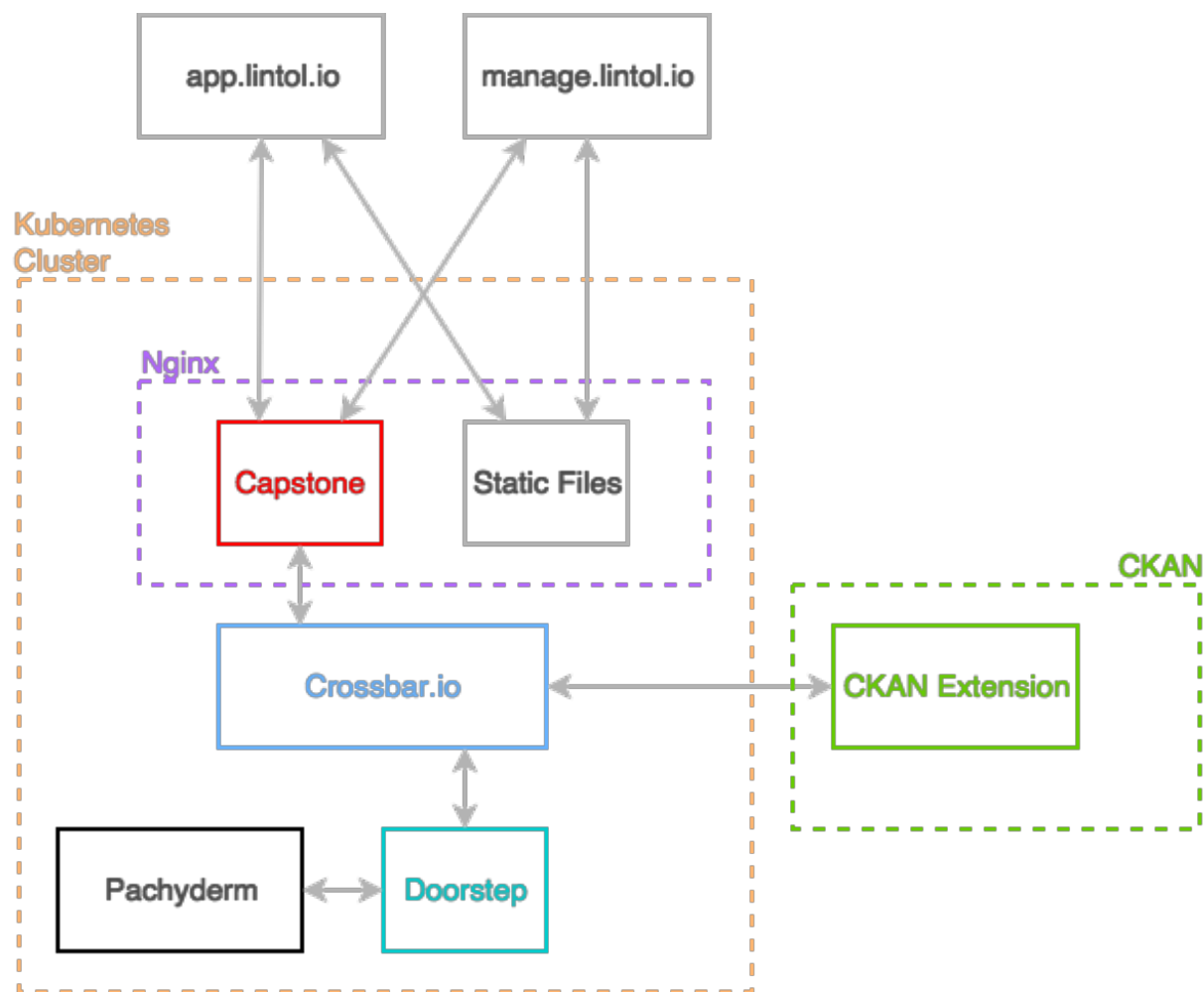
Overview

Components

Lintol has a series of components that have been created using various technologies and programming languages. Distributed components connect via a central component named 'Capstone'. Here is a list of the components, the general function they provide and the technology or programming language used to create them.

- app.lintol.io & manage.lintol.io - Front-end UI - Javascript, Vue
- Crossbar.io - Publish/Subscription and messaging queue
- Doorstep - Data processing job management - python
- CKAN - Open dataset hosting platform - python
- Capstone - Coordinator API - PHP, Laravel

COMPONENTS DIAGRAM



Infrastructure

Components

The infrastructure for Lintol has a number of different components. Each of these components are hosted within a linux server environment. There are a numerous ways of hosting the linux environments required for Lintol. This documentation records and explains the infrastructure used by the Lintol development team.

- AWS EC2 - compute resources
- AWS Route 53 - DNS management
- AWS S3 - data storage
- Kubernetes - container cluster
- Traefik - routing and load balancing


Deployment

TOOLS

Deploying the Lintol infrastructure on the components referenced in the Components section requires a number of deployment and configuration tools.

- AWS CLI
- KubeCtl
- Kops
- Helm - prerequisites [Composer, Docker]

INFRASTRUCTURE DIAGRAM

 Infrastructure diagram needed here

Login and Sign up

There are two main approaches to getting started with Lintol. Lintol can be used as a standalone tool or as in conjunction with a plugin in a platform like CKAN. For both these approaches, Lintol must first be set up through the Lintol management portal.

app.lintol.io

app.lintol.io provides an interface to set up and configure Lintol and how it validates data. It is accessed via a web browser by entering the following url - <http://app.lintol.io>.

The first interface presented to the user is the login interface. The user has the following options:

LOGIN

Enter the user's email address into the 'Email' field Enter the user's password into the 'Password' field Click on the 'Submit' button or press the Enter/Return key

FORGOTTEN PASSWORD

The user can click on 'Forgotten Password' to navigate to the 'Forgotten Password' form

CREATE A NEW USER ACCOUNT

The user can click on 'sign up' to navigate to the user signup form. The user will be required to enter the following pieces of information. - Name - Organisation - Email - Password - Confirm Password The user will then be able to click on the 'Sign Up' button to create a Lintol user account.

Configuration and Administration

Licensing Lintol

Once a user account has been created, the user will need to create a license to use Lintol.

- An account is used by multiple people within Organisation
 - Activate Licence based on domain
 - Pay for Lintol Licence → Generate is Key

Logged In - No Domain (UI)

- Button appears at the top of header to tell them to add domain to activate
- Still have ability to browse "Data Profiles, Processors, Create"
- Once a domain is added any Processors/Data Profiles will be connected to the domain

Logged In - Domain Added

- Still have ability to browse "Data Profiles, Processors, Create"
- Once a domain is added any Processors/Data Profiles will be connected to the domain

Data Profiles

A data profile is an entity that's main purpose is to group together data processors by user preference. A user can create a data profile and then associate data processors with it. Assigning a profile to a data resource will run all the processors against the data resource.

Managing Data Profiles

A list of data profiles can be accessed by clicking on the 'Data Profiles' link in the main menu on the left hand side of the Lintol interface. Each data profile summarises the following information in the list:

- Name
- Description
- Author
- Created date
- Last updated date
- Version
- Unique reference

More details about an individual data profile can be viewed by clicking on [the data profile in the list].

ADD A DATA PROFILE

Adding a data profile can be started by clicking on the "Add a Data Profile" button on the top right off the data profiles page.

Data Processors

Data Processors are the entities that run the validation on the data. A data processor is handed to the job queue along with the data that is going to be validated. Each job then knows which data processor has to be used to validate the data.

Managing Data Processors

A list of data processors can be accessed by clicking on the 'Data Processors' link in the main menu on the left hand side of the Lintol interface. Each data processor summarises the following information in the list:

- Name
- Description
- Created date
- Last updated date
- Unique reference
- Implementations - number of times the processor has been added to a data profile

More details about an individual data processor can be viewed by clicking on [the data processor in the list].

ADD A DATA PROCESSOR

Adding a data processor can be started by clicking on the "Add a Data Processor" button on the top right off the data processors page.

- Sign Up to CKAN instance
- Add a Data Set
 - Choose the Data Profile (Limited to One)
 - Setting Up
 - Technical
 - Enter the Lintol Key into CKAN
 - Non-Technical
 - Email Instruction to technical with Key
 - Technical / Third Part
 - Download/Install
 - Add Lintol Key to the root config file for Lintol
 - Save Config file
 - Add Data Set
 - Choose to Validate Dropdown Lintol Data Profile.

Add Lintol Extension to CKAN

You can install the extension on a CKAN instance as follows.

1. First, ensure you are working within your virtualenv (see prepare-extensions if you are not sure what this means):

```
./home/ubuntu/pyenv/bin/activate
```

2. Install the extension package code using `pip`.

```
pip install -E ~/var/srv/ckan.net/pyenv git+https://github.com/lintol/ckan-plugin
```

The `-E` parameter is for your CKAN Python environment (e.g. `~/var/srv/ckan.net/pyenv`).

The dependency you've installed will appear in the `src/` directory under your Python environment.

3. Add the names of the plugin to the CKAN config file. You can find these in the plugin's `setup.py` file under `ckan.plugins`.

The config plugins variable is in the '[app:main]' section under 'ckan.plugins'. e.g.:

```
[app:main] ckan.plugins = linter
```

If your extension implements multiple different plugin interfaces, separate them with spaces:

```
ckan.plugins = linter myplugin
```

4. If necessary, restart WSGI, which usually means restarting Apache:

```
sudo /etc/init.d/apache2 restart
```

Your extension should now be installed.

Contributing

We love pull requests from everyone. By participating in this project, you agree to abide by the linter [code of conduct]. When contributing to this repository, please first discuss the change you wish to make via issue, email, or any other method with the owners of this repository before making a change.

- Fork the repository
- Clone the repo - `git clone https://github.com/your-username/linter-frontend.git`
- Push to your fork and submit a pull request

At this point you're waiting on us. We like to at least comment on pull requests within three business days (and, typically, one business day). We may suggest some changes or improvements or alternatives.

Some things that will increase the chance that your pull request is accepted:

- Write tests.
- Follow our [style guide][style].
- Write a [good commit message][commit].

Contributor Code of Conduct

Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at [INSERT EMAIL ADDRESS]. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 1.4, available at <https://www.contributor-covenant.org/version/1/4/code-of-conduct.html>

Sample markdown

Subheader 1

SUBHEADER 2

Parameter	Default	Description
param	false	If set to true.

!

warning

i

notice

code

 * list * list

alt text