
kedro-argo

Release 0.0.9

Aug 11, 2020

Contents

1	Kedro-Argo	1
1.1	Installation	3
1.2	Requirements	3
1.3	Usage	3
1.4	Documentation	4
1.5	Development	4
2	Installation	5
3	Usage	7
4	Reference	9
4.1	kedro_argo	9
5	Contributing	11
5.1	Bug reports	11
5.2	Documentation improvements	11
5.3	Feature requests and feedback	11
5.4	Development	12
6	Authors	13
7	Changelog	15
7.1	0.0.9 (2020-08-11)	15
7.2	0.0.8 (2020-04-27)	15
7.3	0.0.7 (2020-03-27)	15
7.4	0.0.5 (2020-03-08)	15
7.5	0.0.4 (2020-03-07)	15
7.6	0.0.0 (2020-03-07)	16
8	Indices and tables	17
	Python Module Index	19
	Index	21

CHAPTER 1

Kedro-Argo

docs	
tests	
package	

Converting kedro pipelines to argo pipelines.

`docs/images/kedro-argo.png`

- Free software: BSD 3-Clause License

1.1 Installation

```
pip install kedro-argo
```

You can also install the in-development version with:

```
pip install https://github.com/nraw/kedro-argo/archive/master.zip
```

1.2 Requirements

- To be used with Kedro, so it's assumed this package is used with a Kedro project.
- Argo CLI is needed for the deployment step. It's also assumed that Argo is already installed on your kubernetes instance.
- You must specify an image name as a parameter. You can generate the image using Kedro-docker.
- As the workflow will be in Argo, which means every step will run in its own container. Therefore, all datasets should be somehow passed between containers or else the pipeline will fail. This means either all datasets should be saved externally (S3, Azure, etc.) or in a shared folder that your deployment would have access to.

1.3 Usage

When installed, argo should be visible under your kedro commands, if you're in a kedro project

```
kedro
```

Then you have two options for obtaining the yaml file, namely via Helm or via ytt.

1.3.1 Helm

```
kedro argo IMAGE_NAME
```

Add this repository to your helm charts:

```
helm repo add kedro-argo https://nraw.github.io/kedro-argo-helm/
```

Then either directly install it by passing the kedro.yaml for input values

```
helm install -f templates/kedro.yaml kedro-argo kedro-argo/kedro-argo
```

Or clone it to your repository and change anything that you would still need:

```
helm pull kedro-argo/kedro-argo --untar
```

1.3.2 ytt

Get the kedro.yaml file by running

```
kedro argo --ytt IMAGE_NAME
```

You can now run:

```
ytt -f templates > argo.yaml
```

or if you prefer in Docker:

```
docker run --rm -it --name ytt -v $(pwd)/templates:/templates gerritk/ytt:latest -f /  
->templates > argo.yaml
```

and finally

```
argo submit --watch argo.yaml
```

1.4 Documentation

<https://kedro-argo.readthedocs.io/>

1.5 Development

To run the all tests run:

```
tox
```

Note, to combine the coverage data from all the tox environments run:

Windows	<pre>set PYTEST_ADDOPTS=--cov-append tox</pre>
Other	<pre>PYTEST_ADDOPTS=--cov-append tox</pre>

CHAPTER 2

Installation

At the command line:

```
pip install kedro-argo
```


CHAPTER 3

Usage

To use kedro-argo in a project:

```
import kedro_argo
```


4.1 kedro_argo

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

5.1 Bug reports

When [reporting a bug](#) please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.2 Documentation improvements

kedro-argo could always use more documentation, whether as part of the official kedro-argo docs, in docstrings, or even on the web in blog posts, articles, and such.

5.3 Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/nraw/kedro-argo/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that code contributions are welcome :)

5.4 Development

To set up *kedro-argo* for local development:

1. Fork *kedro-argo* (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:nraw/kedro-argo.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. When you’re done making changes run all the checks and docs builder with `tox` one command:

```
tox
```

5. Commit your changes and push your branch to GitHub:

```
git add .
git commit -m "Your detailed description of your changes."
git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

5.4.1 Pull Request Guidelines

If you need some code review or feedback while you’re developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`)¹.
2. Update documentation when there’s new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

5.4.2 Tips

To run a subset of tests:

```
tox -e envname -- pytest -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

¹ If you don’t have all the necessary python versions available locally you can rely on Travis - it will run the tests for each change you add in the pull request.
It will be slower though ...

CHAPTER 6

Authors

- Andrej Marsic - nraw.eu

7.1 0.0.9 (2020-08-11)

- Added option for selecting a specific pipeline with `-pipeline` command

7.2 0.0.8 (2020-04-27)

- Changed the `ytt` option to be a flag

7.3 0.0.7 (2020-03-27)

- Changed the default templating option to be Helm instead of `ytt`

7.4 0.0.5 (2020-03-08)

- Dirty names are now transformed to make less likely that symbols break Argo

7.5 0.0.4 (2020-03-07)

- Refactoring and initial adaptation
- Inclusion of tests

7.6 0.0.0 (2020-03-07)

- First release on PyPI.

CHAPTER 8

Indices and tables

- `genindex`
- `modindex`
- `search`

k

kedro_argo, 9

K

kedro_argo (*module*), 9