datalad_container Documentation

Release 0.1.0

DataLad team

Jan 12, 2019

Contents

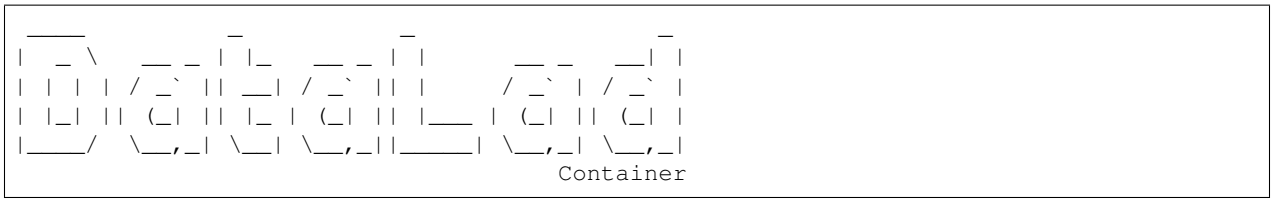
This extension equips DataLad's run/rerun functionality with the ability to transparently execute commands in containerized computational environments. On re-run, DataLad will automatically obtain any required container at the correct version prior execution.

CHAPTER 1

Documentation

- Documentation index
- *Getting started*
- *API reference*

1.1 Change log



This is a high level and scarce summary of the changes between releases. We would recommend to consult log of the DataLad git repository for more details.

1.1.1 0.2.2 (Dec 19, 2018) – The more the merrier

- list/use containers recursively from installed subdatasets
- Allow to specify container by path rather than just by name
- Adding a container from local filesystem will copy it now

1.1.2 0.2.1 (Jul 14, 2018) – Explicit lyrics

- Add support `datalad run --explicit`.

1.1.3 0.2 (Jun 08, 2018) – Docker

- Initial support for adding and running Docker containers.
- Add support `datalad run --sidecar`.
- Simplify storage of `call_fmt` arguments in the Git config, by benefitting from `datalad run` being able to work with single-string compound commmands.

1.1.4 0.1.2 (May 28, 2018) – The docs

- Basic beginner documentation

1.1.5 0.1.1 (May 22, 2018) – The fixes

New features

- Add container images straight from singularity-hub, no need to manually specify `--call-fmt` arguments.

API changes

- Use "name" instead of "label" for referring to a container (e.g. `containers-run -n ...` instead of `containers-run -l`.

Fixes

- Pass relative container path to `datalad run`.
- `containers-run` no longer hides `datalad run` failures.

1.1.6 0.1 (May 19, 2018) – The Release

- Initial release with basic functionality to add, remove, and list containers in a dataset, plus a `run` command wrapper that injects the container image as an input dependency of a command call.

1.2 Acknowledgments

DataLad development is being performed as part of a US-German collaboration in computational neuroscience (CRCNS) project "DataGit: converging catalogues, warehouses, and deployment logistics into a federated 'data distribution'" (Halchenko/Hanke), co-funded by the US National Science Foundation (NSF 1429999) and the German Federal Ministry of Education and Research (BMBF 01GQ1411). Additional support is provided by the German federal state of Saxony-Anhalt and the European Regional Development Fund (ERDF), Project: Center for Behavioral Brain Sciences, Imaging Platform

DataLad is built atop the git-annex software that is being developed and maintained by Joey Hess.

1.3 Getting started

1.3.1 Getting started

The Datalad container extension provides a few commands to register containers with a dataset and use them for execution of arbitray commands. In order to get going quickly, we only need a dataset and a ready-made container. For this demo we will start with a fresh dataset and a demo container from Singularity-Hub.

```
# fresh dataset
datalad create demo
cd demo

# register container straight from Singularity-Hub
datalad containers-add my1st --url shub://datalad/datalad-container:testhelper
```

This will download the container image, add it to the dataset, and record basic information on the container under its name "my1st" in the dataset's configuration at `.datalad/config`.

Now we are all set to use this container for command execution. All it needs is to swap the command *datalad run* with *datalad containers-run*. The command is automatically executed in the registered container and the results (if there are any) will be added to the dataset:

```
datalad containers-run cp /etc/debian_version proof.txt
```

If there is more than one container registered, the desired container needs to be specifed via the `--name` option. Containers do not need to come from Singularity-Hub, but can be local images too. Via the `containers-add --call-fmt` option it is possible to configure how exactly a container is being executed, or which local directories shall be made available to a container.

At the moment there is built-in support for Singularity images, but other container execution systems can be used together with custom helper scripts. Direct support for Docker is under development.

The Datalad container extension provides a few commands to register containers with a dataset and use them for execution of arbitray commands. In order to get going quickly, we only need a dataset and a ready-made container. For this demo we will start with a fresh dataset and a demo container from Singularity-Hub.

```
# fresh dataset
datalad create demo
cd demo

# register container straight from Singularity-Hub
datalad containers-add my1st --url shub://datalad/datalad-container:testhelper
```

This will download the container image, add it to the dataset, and record basic information on the container under its name "my1st" in the dataset's configuration at `.datalad/config`.

Now we are all set to use this container for command execution. All it needs is to swap the command *datalad run* with *datalad containers-run*. The command is automatically executed in the registered container and the results (if there are any) will be added to the dataset:

```
datalad containers-run cp /etc/debian_version proof.txt
```

If there is more than one container registered, the desired container needs to be specifed via the `--name` option. Containers do not need to come from Singularity-Hub, but can be local images too. Via the `containers-add --call-fmt` option it is possible to configure how exactly a container is being executed, or which local directories shall be made available to a container.

At the moment there is built-in support for Singularity images, but other container execution systems can be used together with custom helper scripts. Direct support for Docker is under development.

CHAPTER 2

API Reference

2.1 Command manuals

2.1.1 datalad-containers-add

Synopsis

```
datalad-containers-add [-h] [-u URL] [-d DATASET] [--call-fmt FORMAT] [-i IMAGE] [--
↪update] NAME
```

Description

Add a container to a dataset

Options

NAME

The name to register the container under. This also determines the default location of the container image within the dataset. Constraints: value must be a string

-h, –help, –help-np

show this help message. –help-np forcefully disables the use of a pager for displaying the help message

-u URL, –url URL

A URL (or local path) to get the container image from. If the URL scheme is 'shub://', the command format string will be auto-guessed when –call-fmt is not specified. For the scheme 'dhub://', the rest of the URL will be interpreted as the argument to 'docker pull', the image will be saved to the location specified by NAME, and the call format will be auto-guessed if not given. Constraints: value must be a string [Default: None]

-d *DATASET*, –dataset *DATASET*

specify the dataset to add the container to. If no dataset is given, an attempt is made to identify the dataset based on the current working directory. Constraints: Value must be a Dataset or a valid identifier of a Dataset (e.g. a path) [Default: None]

–call-fmt FORMAT

Command format string indicating how to execute a command in this container, e.g. "singularity exec {img} {cmd}". Where '{img}' is a placeholder for the path to the container image and '{cmd}' is replaced with the desired command. Constraints: value must be a string [Default: None]

-i IMAGE, –image IMAGE

Relative path of the container image within the dataset. If not given, a default location will be determined using the NAME argument. Constraints: value must be a string [Default: None]

–update

Update the existing container for NAME. If no other options are specified, URL will be set to 'updateurl', if configured. If a container with NAME does not already exist, this option is ignored. [Default: False]

Authors

datalad is developed by The DataLad Team and Contributors <team@datalad.org>.

2.1.2 datalad-containers-remove

Synopsis

```
datalad-containers-remove [-h] [-d DATASET] [-i] NAME
```

Description

Remove a known container from a dataset

Options

NAME

name of the container to remove. Constraints: value must be a string

-h, –help, –help-np

show this help message. –help-np forcefully disables the use of a pager for displaying the help message

-d *DATASET*, –dataset *DATASET*

specify the dataset to query. If no dataset is given, an attempt is made to identify the dataset based on the current working directory. Constraints: Value must be a Dataset or a valid identifier of a Dataset (e.g. a path) [Default: None]

-i, –remove-image

if set, remove container image as well. [Default: False]

Authors

datalad is developed by The DataLad Team and Contributors <team@datalad.org>.

2.1.3 datalad-containers-list

Synopsis

```
datalad-containers-list [-h] [-d DATASET]
```

Description

List containers known to a dataset

Options

-h, –help, –help-np

show this help message. –help-np forcefully disables the use of a pager for displaying the help message

-d *DATASET*, –dataset *DATASET*

specify the dataset to query. If no dataset is given, an attempt is made to identify the dataset based on the current working directory. Constraints: Value must be a Dataset or a valid identifier of a Dataset (e.g. a path) [Default: None]

Authors

datalad is developed by The DataLad Team and Contributors <team@datalad.org>.

2.1.4 datalad-containers-run

Synopsis

```
datalad-containers-run [-h] [-n NAME] [-d DATASET] [-i PATH] [-o PATH] [-m MESSAGE] [-
↪-expand WHICH] [--explicit] [--sidecar yes|no] ...
```

Description

Drop-in replacement of 'run' to perform containerized command execution

Container(s) need to be configured beforehand (see containers-add). If only one container is known, it will be selected automatically, otherwise a specific container has to be specified.

A command is generated based on the input arguments such that the container image itself will be recorded as an input dependency of the command execution in the RUN record in the git history.

Options

COMMAND

command for execution.

-h, –help, –help-np

show this help message. –help-np forcefully disables the use of a pager for displaying the help message

-n NAME, –container-name NAME

Specify the name of or a path to a known container to use for execution, in case multiple containers are configured. [Default: None]

-d *DATASET*, –dataset *DATASET*

specify the dataset to record the command results in. An attempt is made to identify the dataset based on the current working directory. If a dataset is given, the command will be executed in the root directory of this dataset. Constraints: Value must be a Dataset or a valid identifier of a Dataset (e.g. a path) [Default: None]

-i PATH, –input PATH

A dependency for the run. Before running the command, the content of this file will be retrieved. A value of "." means "run datalad get .". The value can also be a glob. This option can be given more than once. [Default: None]

-o PATH, –output PATH

Prepare this file to be an output file of the command. A value of "." means "run datalad unlock ." (and will fail if some content isn't present). For any other value, if the content of this file is present, unlock the file. Otherwise, remove it. The value can also be a glob. This option can be given more than once. [Default: None]

-m MESSAGE, –message MESSAGE

a description of the state or the changes made to a dataset. Constraints: value must be a string [Default: None]

–expand WHICH

Expand globs when storing inputs and/or outputs in the commit message. Constraints: value must be NONE, or value must be one of ('inputs', 'outputs', 'both') [Default: None]

–explicit

Consider the specification of inputs and outputs to be explicit. Don't warn if the repository is dirty, and only save modifications to the listed outputs. [Default: False]

–sidecar yes|no

By default, the configuration variable 'datalad.run.record-sidecar' determines whether a record with information on a command's execution is placed into a separate record file instead of the commit message (default: off). This option can be used to override the configured behavior on a case-by-case basis. Sidecar files are placed into the dataset's '.datalad/runinfo' directory (customizable via the 'datalad.run.record-directory' configuration variable). Constraints: value must be NONE, or value must be convertible to type bool [Default: None]

Authors

datalad is developed by The DataLad Team and Contributors <team@datalad.org>.

2.2 Python API

containers_add	Add a container environment to a dataset
containers_remove	Remove a container environment from a dataset
containers_list	List known container environments of a dataset
containers_run	Drop-in replacement for *datalad run* for command execution in a container

2.2.1 datalad_container.containers_add

Add a container environment to a dataset

class datalad_container.containers_add.**ContainersAdd**
Bases: datalad.interface.base.Interface

Add a container to a dataset

2.2.2 datalad_container.containers_remove

Remove a container environment from a dataset

class datalad_container.containers_remove.**ContainersRemove**
Bases: datalad.interface.base.Interface

Remove a known container from a dataset

2.2.3 datalad_container.containers_list

List known container environments of a dataset

class datalad_container.containers_list.**ContainersList**
Bases: datalad.interface.base.Interface

List containers known to a dataset

2.2.4 datalad_container.containers_run

Drop-in replacement for *datalad run* for command execution in a container

class datalad_container.containers_run.**ContainersRun**
Bases: datalad.interface.base.Interface

Drop-in replacement of 'run' to perform containerized command execution

Container(s) need to be configured beforehand (see containers-add). If only one container is known, it will be selected automatically, otherwise a specific container has to be specified.

A command is generated based on the input arguments such that the container image itself will be recorded as an input dependency of the command execution in the *run* record in the git history.

Python Module Index

d

Index

C

D