getlino

Release 21.3.4

unknown

Apr 30, 2021

CONTENTS

`getlino` is a command-line tool for installing Lino in different environments.

Note: If you **just want to install** Lino, then this repository is **not for you**. You want to read the Lino book instead. This repository is for people who want to help with developing the tool that you use to install Lino.

- Project homepage: https://github.com/lino-framework/getlino
- Documentation: https://getlino.lino-framework.org
- Test results: https://travis-ci.com/github/lino-framework/getlino

CHAPTER

ONE

USING DOCKER TO TEST GETLINO

Testing getlino is quite different from testing Lino itself in that getlino is a tool that modifies the system configuration of a Linux machine. You obviously don't want it to do this on the computer you are using for development.

1.1 Quick introduction to Docker

A Docker **container** is like a virtual machine, but it stores only the things that you changed after installing the **image**. A Docker **image** contains a whole operating system:.

Running `inv prep` in the getlino repository will build a set of Docker images needed for running the getlino test suite. You don't need to run it before every test run, only once after each modification in the getlino source code.

- `debian_updated` = a virgin Debian buster + apt upgrade + create a user "lino"
- `debian_with_getlino` = *debian_updated* + add the current getlino source files
- `ubuntu_updated` : same as *debian_updated* but with Ubuntu
- `ubuntu_with_getlino` : same as *debian_with_getlino* but with Ubuntu

You can see the available docker images on your computer:

```
$ docker image ls
REPOSITORY            TAG                 IMAGE ID            CREATED              SIZE
ubuntu_with_getlino   latest              509d56b09981        3 hours ago         ␣
 ↪513MB
debian_with_getlino   latest              a96d832cb84c        3 hours ago         ␣
 ↪598MB
ubuntu_updated        latest              4171a574c2d7        4 days ago          ␣
 ↪513MB
debian_updated        latest              f29ed368ec5d        4 days ago          ␣
 ↪598MB
ubuntu                bionic              d27b9ffc5667        2 weeks ago          64.
 ↪2MB
debian                buster              1b686a95ddbf        6 weeks ago         ␣
 ↪114MB
```

You can run an interactive session in a docker container using one of those images:

```
$ docker run -it debian_with_getlino /bin/bash
lino@97621d803236:/src$
```

The effect of this is that you are now inside a bash session on your container. Feel free to play around:

```
lino@97621d803236:/src$ pwd
/src
lino@97621d803236:/src$ ls -al
total 16
drwxr-xr-x 1 lino lino 4096 Jul 27 09:50 .
drwxr-xr-x 1 root root 4096 Jul 27 11:24 ..
drwxr-xr-x 4 root root 4096 Jul 27 09:50 getlino
-rw-rw-r-- 1 root root  260 Jul 10 10:02 setup.py
lino@97621d803236:/src$ echo "Hello, world!" > hello.txt
```

Now hit `Ctrl-D` to terminate your bash session. This will put your container into the state “exited”. You can see the status of all your containers by saying:

```
$ docker ps -a
CONTAINER ID        IMAGE                  COMMAND             CREATED             ␣
↪STATUS                  PORTS               NAMES
97621d803236        debian_with_getlino    "/bin/bash"         About a minute ago  ␣
↪Exited (0) 5 seconds ago                    sharp_austin
```

Copy the name of your container (*97621d803236* in this example, but yours is likely to be different) to your clipboard. Now let’s say you want to continue working on this container:

```
$ docker start -i 97621d803236
lino@97621d803236:/src$ cat hello.txt
Hello, world!
lino@97621d803236:/src$  # hit [Ctrl-D] to exit
```

As you can see, the `hello.txt` from your previous session is still there.

You can start a container in “detached mode”, that is, without attaching it to your terminal. The `docker start` command returns you immediately to the shell prompt and the container continues running in background:

```
$ docker start 97621d803236
```

You can now run one bash command at a time from the command line:

```
$ docker exec 97621d803236 /bin/bash -c "cat hello.txt"
Hello, world!
$
```

We can verify that the container is still running in the background:

```
$ docker ps -a
CONTAINER ID  IMAGE                 COMMAND       CREATED       STATUS          PORTS ␣
↪NAMES
97621d803236  debian_with_getlino  "/bin/bash"    2 hours ago   Up 10 seconds         ␣
↪sharp_austin
```

Let’s tidy up and remove our container:

```
$ docker container rm 97621d803236
Error response from daemon: You cannot remove a running container
97621d803236e46b66917aae8bc6fb01ea3ab3f8749e374d33a818516c833509.
Stop the container before attempting removal or force remove
```

Yes, we started the container in detached mode, it would run forever if we don’t stop it:

```
$ docker container stop 97621d803236
97621d803236
```

Now we can remove it:

```
$ docker container rm 97621d803236
```

What we saw here is basically all we do in our test suite. Let's have a look at the file `tests/test_docker.py`

> $ docker run –publish 8000:8080 –detach –name mycont getlino_debian

1.2 Docker uses much disk space

How to see how much disk space docker is using on your computer:

```
$ docker system df
TYPE                TOTAL               ACTIVE              SIZE               ␣
↪RECLAIMABLE
Images              34                  5                   5.1GB               5.1GB␣
↪(99%)
Containers          11                  2                   17GB                14.
↪13GB (83%)
Local Volumes       0                   0                   0B                  0B
Build Cache         0                   0                   0B                  0B
```

To get more details, you can also run:

```
$ docker system df -v
```

From time to time I tidy up and remove all rebuildable containers:

```
$ docker system prune
WARNING! This will remove:
  - all stopped containers
  - all networks not used by at least one container
  - all dangling images
  - all dangling build cache

Are you sure you want to continue? [y/N] y
Deleted Containers:
cdd408dc0ee130d4498c82f0eed6609445b3ae290ef21c7739ef29ceca99fbd4
493ae1128f25bc144598661eaf854de527cdc7b4795ba1a34f9e46a0aa852012
48f9d5220778b8efd7db4bb041659b9b058f993e234e770e803e4cbeb18e4124
...
Total reclaimed space: 27.53GB
```

Sources consulted:

- https://docker-curriculum.com/
- https://stackoverflow.com/questions/21928691/how-to-continue-a-docker-container-which-has-exited

CHAPTER

TWO

API

—

(This module’s source code is available here.)

CHAPTER THREE

MISCELLANEOUS

```
>>> from getlino.utils import perm2text
>>> print(perm2text(0o3777))
rwxrwsrwt
>>> print(perm2text(0o2775))
rwxrwsr-x
>>> print(perm2text(1533))
rwxrwsr-x
>>> print(perm2text(509))
rwxrwxr-x
>>> print(perm2text(436))
rw-rw-r--
```

```
>>> print(perm2text(1))
--------x
```

```
>>> print(perm2text(0o123456))
Traceback (most recent call last):
...
Exception: value must be less than 0o7777
```

CHAPTER

FOUR

CHANGES IN *GETLINO*

4.1 2021-04-27

Added a new command `getlino list`.

4.2 2021-04-12

Updated the KNOWN_REPOS in `getlino.utils`: the following repositories have moved to GitLab: lino, xl, noi, cosi, voga, avanti, welfare, weleup, welcht

4.3 2021-03-16

You can now run `getlino configure` as root with `--clone` and without `--shared-env`, provided that you activated a virtualenv before calling getlino. In that case getlino will install clones into the current virtualenv and store the current virtualenv in the system-wide config file. Miscellaneous optimizations and bugfixes when setting up a demo server.

Release to PYPI.

More minor bugfixes: healthcheck.sh wasn't set to executable. configure failed when –https was given but no –web-server

4.4 2021-03-14

getlino forgot to run **`sudo ln -s /snap/bin/certbot /usr/bin/certbot`** when installing certbot.

4.5 2021-03-11

When *–https* is specified but certbot is not installed, getlino now installs it using snapd (no longer using wget and apt-get)

4.6 2021-03-08

Release 21.3.0 to PyPI.

4.7 2021-03-05

Optimization: When getlino asks to change file permissions, it now formats them as "rwx" style strings instead of decimal integers.

4.8 2021-03-02

Fixed #3998 (getlino says Invalid value for '–web-server': invalid choice: . (choose from nginx, apache)). Reactivated UbuntuDockerTest in test suite. getlino now upgrades the system not only when you are root but also when you are member of sudo. It no longer asks "Upgrade the system?" as a separate question, only when it actually wants to install packages (and when you are either root or sudoer).

4.9 2021-02-13

Fixed a bug in the generated `make_snapshot.sh` file: it was testing *if [-f media/uploads]* instead of *if [-d media/uploads]*, as a result the uploads were never included in the snapshot.

4.10 2021-02-12

Fixed a problem when using apache config: certbot made a copy of the 80 conf file, but failed to copy the WSGI* directives. Now getlino creates a 443 site with snakeoil certificate already from the beginning. Added *uploads* to `make_snapshots.py`.

Released version 21.2.1 to PyPI.

4.11 2021-02-10

Changed the number of nginx worker processes in the uwsgi.ini script from 2 to 1 as every worker process immobilizes about 5 to 6% of 2GB of RAM even when nobody is using the site.

Added support for apache web server. Until now, getlino always installed nginx (when running as root). Now `getlino configure` has a new option *–web-server*, which can be "nginx", "apache" or empty. Changed behaviour: When not given, getlino will not setup any web server configuration, even when running as root.

The test suite now also tests for "ERROR" (not just "Error") in the output of `healthcheck.sh`. Increased the wait time for supervisor to restart from 10 to 20 seconds because with only 10 seconds it failed once.

Released version 21.2.0 to PyPI.

4.12 2021-02-08

Fixed several minor bugs in *getlino startsite*: Fixed a typo bug that caused `--https` to fail at the last step (when calling certbot for the new subdomain). Some config files were generated with a leading newline, and the make_snapshot cron job even with leading blanks on every line. The `nginx.conf` file was still pointing `/static/` to a directory static, but the new default name for this is `static_root`.

4.13 2020-09-23

getlino configure now again installs (or instructs to install) apt packages libldap2-dev and libsasl2-dev, which are –as it seems– required for django_auth_ldap.

4.14 2020-09-21

When invoking getlino configure without sudo, it no longer asks for *–usergroup*.

Released getlino 20.9.0 to PyPI.

4.15 2020-09-02

getlino can now install certbot either using the Debian packager or using certbot-auto. getlino now supports calling `Installer.run_apt_install()` more than once during an installation session.

4.16 2020-08-27

`getlino startsite` has now the database options (db-engine, db-user etc) so that you can override them per site without needing to run getlino configure. Added a new choice `lino.projects.std` for the *APPNAME* argument of `getlino startsite`. Fixed a warning `bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)` during test run.

4.17 2020-07-29

Released getlino 20.7.5 to PyPI. With a few bugfixes.

4.18 2020-07-20

Released getlino 20.7.3 to PyPI. After some subtle changes.

Released getlino 20.7.4 to PyPI. After some more subtle changes.

4.19 2020-07-19

Released getlino 20.7.2 to PyPI.

4.20 2020-07-17

getlino failed on Python 3.8 because it was using the deprecated `platform.dist`. Now it uses `distro`.

getlino now shows its version.

Released getlino 20.7.1 to PyPI.

local-prefix was root_only, which caused configure to fail when not run as root.

4.21 2020-07-16

getlino failed on Windows because the grp module is not available there. Now we simply skip the group ownership check when running on Windows.

Reorganized the Docker files.

Released getlino 20.7.0 to PyPI.

4.22 2020-06-23

`getlino startsite` now creates a daily cron job that runs `make_snapshot.sh`.

4.23 2020-05-14

Remove useless command to copy "mysql_config" file for MariaDB.

4.24 2020-05-14

Bugfix : `getlino configure` without *–clone*, the *–shared-env* now defaults to an empty string.

4.25 2020-05-04

`getlino configure` now defaults `--shared-env` to the current `VIRTUAL_ENV` only when `--clone` was given, not always. And it creates the `repos_base` only then.

4.26 2020-04-07

Fix typo with apt_packages of DbEngine. Release 20.4.5 version to PyPI.

4.27 2020-04-03

Remove certbot for www domain Release 20.4.3 version to PyPI.

4.28 2020-04-03

Update the virualenv usage. Fix issue with installing mariadb for debian distribution instead of mysql

4.29 2020-01-03

Add the ciao projet to KNOWN_REPOS. Add 'sudo' to the certbot command.

Fix some issues with creating user and database with the `getlino startsite` .

Released version 20.1 to PyPI Released version 20.1.1 to PyPI

No need for 'sudo' for certbot command

Released version 20.1.2 to PyPI

4.30 2019-11-23

When the user is not root , the `getlino startsite` command doesn't create the `make_snapshot.sh` and the directory *nginx*.

4.31 2019-11-09

The `pull.sh` script generated by `getlino configure` into the shared virtualenv was still using the project_dir.

4.32 2019-11-07

The `--https` option was appending directly to the main `/etc/crontab` file. Fixed.

Fixed some minor bugs. For example the `--redis` option was ignored when not running as root.

Released 19.11.0 to PyPI.

4.33 2019-11-06

getlino configure --db-user option now creates the shared database user. The items of getlino.utils.DB_ENGINES are no longer named tuples but real objects with methods.

4.34 2019-10-25

getlino no longer depends on cookiecutter. The separate cookiecutter-startsite repository is no longer used because all templates are now below getlino/templates.

Released version 19.10.6 to PyPI (versions 19.10.3 to 19.10.5 are broken versions, don't use them).

4.35 2019-10-08

Released version 19.10.0 to PyPI.

Fixed some bugs: Running **getlino configure** without --db-port caused an error Invalid value for "--db-port": invalid choice: . (choose from 5432, 3306, 0). The --db-port option is no longer a choice (it is not limited to these values). The --clone option sometimes had *True* as default value when it shouldn't.

Released version 19.10.1 to PyPI.

Found another bug: with --clone, getlino didn't clone the repositories using their nickname, which later caused failures when trying to install them.

Released version 19.10.2 to PyPI.

Fixed some more bugs: Running **getlino configure** without --db-port caused an error Invalid value for "--db-port": invalid choice: . (choose from 5432, 3306, 0). The --db-port option is not a choice (it must not limited to these values). The --clone option sometimes had *True* as default value.

4.36 2019-10-03

The --db-port shows the default ports of databases.

4.37 2019-09-19

The --clone option installs all contributor repositories, i.e. those required to build the book. Some repositories were still missing. Fixed. Also separated the sequence of resulting actions: first run "git clone" for all repos, then "pip install -e".

4.38 2019-09-18

Optimized behaviour when running as non-root: The default value for `--devtools` is now *True* in that case. `--db-engine` had a wrong default value "sqlite" (must be "sqlite3"), getlino tried to create the directories given by `--log-base` and `--backups-base` (which failed because not running as root).

4.39 2019-09-14

When running as root, `getlino configure` now also creates empty directories for `--log-base` and `--backups-base` and sets their permissions.

4.40 2019-09-12

When running as root, getlino now also installs the *build-essential* Debian package because this is maybe needed for installing Python extensions.

getlino didn't set the group owner in some cases (e.g. the lino_local directory and a project's virtualenv).

`getlino configure` now also creates a `~/.bash_aliases` file. But only when you aren't running as root. After running `getlino configure` as root, you may want to run it once more without being root to create a `.bash_aliases` file to your home directory.

Released getlino 19.9.6 to PyPI, immediately followed by a bug-fix release 19.9.7

4.41 2019-09-09

Added more demo projects to be used by the getlino configure command. Released getlino 19.9.5.

4.42 2019-09-08

Renamed the `--contrib` option to `--clone` because it is also used when configuring a demo server. It means "clone all known repositories to the –repos-base and install them to the shared-env using `pip -e`." This change requires that you run `getlino configure` once after upgrade, or manually edit your getlino config file.

The configure command now supports `--clone` without specifying a –repos-base. In that case it uses the *repositories* subdir of the shared-env.

New option `--shared-env` for startsite. When used with startsite, it overrides the value specified during configure.

Released getlino 19.9.4 on PyPI.

4.43 2019-09-07

Released getlino 19.9.2 on PyPI, followed by a bugfix release 19.9.3.

4.44 2019-09-02

Also write logrotate config file for supervisor.

Changed some default values in `getlino configure`: The default value for `--clone` was wrong : when running as root, it is *not* a contributor environment. `--shared-env` and `--repos-base` are now empty when `VIRTUAL_ENV` is not set. And `--db-engine` is now mysql when running as root.

Released getlino 19.9.0 to PyPI, followed by a bugfix release 19.9.1.

4.45 2019-08-27

Released getlino 19.8.1 on PyPI.

4.46 2019-08-01

Released getlino 19.8.0 on PyPI.

4.47 2019-07-30

Added a first meaningful unit test (`test_docker_prod`).

CHAPTER

FIVE

INSTALLING GETLINO

5.1 Install

The easiest way to install getlino is via pip:

```
$ pip install getlino
```

To update your getlino to the newest version simply run:

```
$ pip install -U getlino`
```

5.2 More options

You can optionally use the development version:

```
$ pip uninstall getlino
$ pip install -e git+https://github.com/lino-framework/getlino.git#egg=getlino
```

You may install your own local clone of getlino:

```
$ cd ~/lino/env/repositories
$ git clone git@github.com:lino-framework/getlino.git
$ pip install -e getlino
```

In that last use case don’t forget to manually add getlino to your `./atelier/config.py` file.

CHAPTER SIX

USAGE

Content moved to getlino : the Lino installer.

PYTHON MODULE INDEX

INDEX