
Resen-core

Release v2020.2.0

Jul 28, 2021

Contents

1	resen-core	1
1.1	Documentation	1
2	Usage	3
2.1	Tutorials	3
3	Python Packages Available in Resen-core	5
3.1	Python Packages included	5
4	Notes for Developers	7
4.1	Resen-core images	7
4.2	Building a resen-core image	7
4.2.1	Resen-base	8
4.2.2	resen-core Dockerfile helpers	8
4.3	Using a resen-core image without the resen tool	8
5	Changelog	9
5.1	2021.1.0 (2021-07-28)	9
5.2	2020.2.0 (2020-11-11)	9
5.3	2020.1.0 (2020-06-15)	9
5.4	2019.1.0 (2019-10-24)	10

CHAPTER 1

resen-core

`resen-core` is the core docker image used by `resen`. `resen` ([REproducible Software ENvironment](#)), is a tool that enables reproducible scientific data analysis, built using Python and Docker. `resen-core` is built upon an Ubuntu Linux image with Python, Jupyter, and a stack of commonly used Geospace research software libraries and packages pre-installed.

Users who are only interested in using Resen for their scientific research and not developing the system should refer to the [resen documentation](#). The files and documentation included in this repository are intended primarily for developers and advanced users who would like to customize their `resen-core` docker image.

`resen` and `resen-core` were developed under the InGeo project, currently supported by the National Science Foundation's Cyberinfrastructure for Sustained Scientific Innovation (CSSI) program (Grant #1835573). For more information about the InGeo project, please visit the [InGeo website](#).

1.1 Documentation

Complete documentation is available at resen-core.readthedocs.io.

Specific versions of `resen-core` are available through `resen`. When creating a `bucket` the user is asked to select a specific `resen-core` version to base their `bucket` on.

2.1 Tutorials

A number of `tutorials` are available that illustrate how to get started working in a `resen` bucket and use some of the common tools that are available. Instructions for how to access these tutorials in a bucket are available [here](#).

Python Packages Available in Resen-core

This version of resen-core includes one python virtual environment, py38, based on python3.8. Below are the packages available in that environment.

3.1 Python Packages included

Table 1: py38 environment packages

Package	Version	Application	Source
aacgm2	2.6.2	AACGM magnetic coordinate system	https://pypi.org/project/aacgm2/
apexpy	1.1.0	Apex magnetic coordinate system	https://pypi.org/project/apexpy/
astropy	4.2.1.post1	Packages for use in astronomy	https://www.astropy.org/
basemap	1.2.1	Mapping	https://matplotlib.org/basemap/
bokeh	2.3.2	Interactive visualization library	https://bokeh.pydata.org/en/latest/
cartopy	0.19.0	Mapping	https://scitools.org.uk/cartopy/docs/latest/
cdflib	0.3.20	CDF data format	https://pypi.org/project/cdflib/
citationhelp	0.3	Tracking software to cite	https://pypi.org/project/citationhelper/
ephem	3.7.7.1	High-precision astronomy computations	https://rhodesmill.org/pyephem/
h5py	3.2.1	HDF5 binary data format	https://www.h5py.org/
igrf	13.0.0	IGRF empirical model	https://pypi.org/project/igrf/
iri2016	1.11.1	IRI empirical model	https://pypi.org/project/iri2016/
madrigalweb	1.1.12	Accessing the Madrigal database	http://cedar.openmadrigal.org/docs/name/rr_python.html
mangopy	0.1	MANGO data analysis	https://github.com/astib/MANGO
matplotlib	3.4.2	Basic plotting	https://matplotlib.org/
msise00	1.10.0	NRL MSISE-00 empirical model	https://pypi.org/project/msise00/
netcdf4	1.5.6	netCDF4 data format	https://unidata.github.io/netcdf4-python/netCDF4/index.html
numpy	1.20.3	Numerical array handling	http://www.numpy.org/
OMMBV	0.5.4	Magnetic basis vectors	https://pypi.org/project/OMMBV/

Continued on next page

Table 1 – continued from previous page

Package	Version	Application	Source
pandas	1.2.4	Data analysis	https://pandas.pydata.org/
plasma	0.6.0	Package for plasma research	https://www.plasmapy.org/
pydarn	2.1	SuperDARN data analysis and visualization	https://pydarn.readthedocs.io/en/master/
pyglow	0.0.0 (Jan 4 2021)	Upper atmosphere climatological models	https://github.com/timduly4/pyglow/
pymap3d	2.7.0	Coordinate transforms and geodesy functions	https://geospace-code.github.io/pymap3d/
pysat	3.0.0	Space physics data access	https://pypi.org/project/pysat/
scipy	1.6.3	Advanced mathematical operations	https://www.scipy.org/
sciunit2	0.4.post58 (2021-05-06)	Python 2 and 3 compatible reusable research objects	https://pypi.org/project/sciunit2/
seaborn	0.11.1	Statistical data visualization	https://seaborn.pydata.org/
sgp4	2.19	Propagation of satellite TLEs	https://pypi.org/project/sgp4/
skyfield	1.39	High-precision astronomy computations	https://rhodesmill.org/skyfield/
spacepy	0.2.2	Data analysis tools for space-based datasets	https://pythonhosted.org/SpacePy/
sqlalchemy	1.4.15	Database tool kit	https://www.sqlalchemy.org/
sunpy	3.0.0	Open-source solar data analysis environment	https://docs.sunpy.org
sympy	1.8	Symbolic computation	https://www.sympy.org/en/index.html
tables	3.6.1	HDF5 binary data format	https://pypi.org/project/tables/
viresclient	0.8.0	Interface to access ESA's Swarm data and models	https://pypi.org/project/viresclient/
visuamir	2.0.3	Read and visualize AMISR data	https://github.com/asreimer/visuamir
xarray	0.18.2	Labeled multi-dimensional arrays	http://xarray.pydata.org/en/stable/

Notes to create, build, and test resen-core images.

4.1 Resen-core images

Alternatively to accessing resen-cores images through [resen](#), the images can be pulled from [earthcubeingeo](#) on [dock-erhub](#) (this is how [resen](#) obtains the selected resen-core image). Once the resen-core image has been pulled into the user's system it will be readily available and not require downloading in the future. To pull a resen-core image from [earthcubeingeo](#) the following [docker](#) command can be used:

```
$ docker pull earthcubeingeo/resen-core:2020.2.0
```

After issuing the command, docker starts downloading the layers contained in the image. When the process finishes the image will be available in the user's system:

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED
↪ SIZE			
earthcubeingeo/resen-core	2020.2.0	b1f1c9013924	1 day ago
↪ 5.25GB			

4.2 Building a resen-core image

The sources for building a resen-core image are in the [resen_core](#) *GitHub* repository. The *Dockerfile* for the resen-core image can be found inside the resen-core folder in the repository. To build the image from the resen-core folder run:

```
$ docker build -t resencoretest .
```

After a successful build, which can take some time, the newly created image should be available in the user's docker list:

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
resencoretest	latest	5431treww4r12	2 hours ago	5.38GB

4.2.1 Resen-base

The resen-core images are based on the resen-base docker image, whos *Dockerfile* is located inside the resen-base folder in the [resen_core GitHub](#) repository. The resen-base image is in turn based on the ubuntu:20.04 docker image found in [ubuntu Docker Official Images](#).

Building a resen-base image

The sources for building a resen-base image are in the [resen_core GitHub](#) repository. The *Dockerfile* for the resen-core image can be found inside the resen-base folder in the repository. To build the image from inside the resen-base folder run:

```
$ docker build -t resenbasetest .
```

To use the newly generated resen-base image and use it in resen-core, the *resen-core/Dockerfile* needs to be modified so that it uses *resenbasetest* instead of *earthcubeingeo/resen-base:2020.2.0*

```
FROM resenbasetest:latest
```

4.2.2 resen-core Dockerfile helpers

Resen-core uses additional files (helpers) that are called as part of the instructions in the *Dockerfile*. The helpers are located inside the folder *resen-core/resources/helpers*:

```
- install_CDF.sh
- setup_basemap.sh
- setup_py38_env.sh
- setup_pyglow.sh
```

4.3 Using a resen-core image without the resen tool

There might be times when there is the need to use a resen-core image without the [resen](#) tool, e.g. when a new image is being created and has not been integrated in the [resen](#) tool. To proceed you need [docker](#) installed in your system and enough resources allocated for compilation. The following command will start jupyter lab based on the resen-core image that was pulled previously, i.e. *earthcubeingeo/resen-core:2020.2.0*

```
$ docker run --name a_container_name -it -p XXXX:XXXX earthcubeingeo/resen-core:2020.
↪2.0 /bin/bash -c 'source ~/envs/py38/bin/activate && jupyter lab --no-browser --ip_
↪0.0.0.0 --port XXXX --NotebookApp.token=SOMETOKENWORD --KernelSpecManager.ensure_
↪native_kernel=False'
```

where XXXX is the port to be used for *jupyterlab*.

5.1 2021.1.0 (2021-07-28)

- Add empirical models IGRF, IRI2016, MSISE00
- Upgrade the versions of the installed packages in py38
- Added pysat, xarray, and ommbv
- Updating to jupyterlab3, using prebuilt extensions that don't need npm and nodejs
- Update documetation to give a more detailed description of resen-core

5.2 2020.2.0 (2020-11-11)

- Upgrade from ubuntu 18.04 to ubuntu 20.04 as base image for resen-base
- Upgrade to python 3.8 and deprecate python 2 (py38 environment)
- Removed davitpy (depends on python 2 and is superseded by pydarn)
- Upgraded sciunit to python 3 version
- Add packages sunpy, pyephem, skyfield, and seaborn
- Upgrade the versions of the installed packages in py38
- Update tutorials
- Update docs

5.3 2020.1.0 (2020-06-15)

- Add plasmapy, pydarn, viresclient

- Add visuamir to py27 and py36 setup scripts
- Removed davitpy, was deprecated by developers
- Update py36 packages with latest versions
- Add %pylab widget capability
- Add citation helper utility
- Updated resen-base, tracking ubuntu 18.04:20200403 image

5.4 2019.1.0 (2019-10-24)

- Initial release.