
c4cds-wps Documentation

Release 0.2.0

Carsten Ehbrecht

Sep 12, 2019

CONTENTS:

1 Credits	3
2 Indices and tables	9

c4cds-wps (the bird) *c4cds-wps is a bird ...*

A WPS Compute Service for the [Copernicus Climate Data Store](#)

- Free software: Apache Software License 2.0
- Documentation: <https://c4cds-wps.readthedocs.io>.

CREDITS

This package was created with [Cookiecutter](#) and the [bird-house/cookiecutter-birdhouse](#) project template.

1.1 Installation

- *Install from Conda*
- *Install from GitHub*
- *Start c4cds-wps PyWPS service*
- *Run c4cds-wps as Docker container*
- *Use Ansible to deploy c4cds-wps on your System*

1.1.1 Install from Conda

Warning: TODO: Prepare Conda package.

1.1.2 Install from GitHub

Check out code from the c4cds-wps GitHub repo and start the installation:

```
$ git clone https://github.com/cp4cds/c4cds-wps.git
$ cd c4cds
```

Create Conda environment named *c4cds*:

```
$ conda env create -f environment.yml
$ source activate c4cds
```

Install c4cds-wps app:

```
$ pip install -e .
OR
make install
```

For development you can use this command:

```
$ pip install -e .[dev]
OR
$ make develop
```

1.1.3 Start c4cds-wps PyWPS service

After successful installation you can start the service using the c4cds command-line.

```
$ c4cds --help # show help
$ c4cds start # start service with default configuration

OR

$ c4cds start --daemon # start service as daemon
loading configuration
forked process id: 42
```

The deployed WPS service is by default available on:

<http://localhost:5000/wps?service=WPS&version=1.0.0&request=GetCapabilities>.

Note: Remember the process ID (PID) so you can stop the service with `kill PID`.

You can find which process uses a given port using the following command (here for port 5000):

```
$ netstat -nlp | grep :5000
```

Check the log files for errors:

```
$ tail -f pywps.log
```

... or do it the lazy way

You can also use the Makefile to start and stop the service:

```
$ make start
$ make status
$ tail -f pywps.log
$ make stop
```

1.1.4 Run c4cds-wps as Docker container

You can also run c4cds-wps as a Docker container.

Warning: TODO: Describe Docker container support.

1.1.5 Use Ansible to deploy c4cds-wps on your System

Use the [Ansible](#) playbook for PyWPS to deploy c4cds-wps on your system. Here we show an example for remote deployment.

Get the playbook:

```
$ git clone https://github.com/bird-house/ansible-wps-playbook.git
$ cd ansible-wps-playbook
# install roles
$ ansible-galaxy -p roles -r requirements.yml install
```

Edit config:

```
$ cp etc/sample-emu.yml custom.yml
$ vim custom.yml
```

Make sure to configure the extra parameters for the data archive:

```
---
wps_user: wps
wps_group: wps
wps_services:
  - name: c4cds
    hostname: wpsdemo
    port: 80
    extra_config: |
      [data]
      c3s_cmip5_archive_root = /data/c3s-cmip5/output1
      cordex_archive_root = /data/cordex/output
```

Add an inventory file for remote deployment:

```
$ vim wpsdemo.cfg
$ cat wpsdemo.cfg
wpsdemo ansible_ssh_user=ansible
```

Run ansible for remote deployment:

```
$ ansible-playbook --ask-sudo-pass -i wpsdemo.cfg playbook.yml
```

1.2 Configuration

Warning: Please read the PyWPS [documentation](#) to find details about possible configuration options.

1.2.1 Command-line options

You can overwrite the default [PyWPS](#) configuration by using command-line options. See the c4cds-wps help which options are available:

```
$ c4cds start --help
--hostname HOSTNAME      hostname in PyWPS configuration.
--port PORT              port in PyWPS configuration.
```

Start service with different hostname and port:

```
$ c4cds start --hostname localhost --port 5001
```

1.2.2 Use a custom configuration file

You can overwrite the default [PyWPS](#) configuration by providing your own PyWPS configuration file (just modify the options you want to change). Use one of the existing `sample-*.cfg` files as example and copy them to `etc/custom.cfg`.

For example change the hostname (*demo.org*) and logging level:

```
$ cd c4cds
$ vim etc/custom.cfg
$ cat etc/custom.cfg
[server]
url = http://demo.org:5000/wps
outputurl = http://demo.org:5000/outputs

[logging]
level = DEBUG

[data]
c3s_cmip5_archive_root = /data/c3s-cmip5/output1
cordex_archive_root = /data/cordex/output
```

Note: You need to configure the path to the local data archives for C3S_CMIP5 and CORDEX.

Start the service with your custom configuration:

```
# start the service with this configuration
$ c4cds start -c etc/custom.cfg
```

1.3 Developer Guide

- *Building the docs*
- *Running tests*
- *Run tests the lazy way*
- *Prepare a release*
- *Bump a new version*

Warning: To create new processes look at examples in [Emu](#).

1.3.1 Building the docs

First install dependencies for the documentation:

```
$ make develop
```

Run the Sphinx docs generator:

```
$ make docs
```

1.3.2 Running tests

Run tests using `pytest`.

First activate the `c4cds` Conda environment and install `pytest`.

```
$ source activate c4cds
$ pip install -r requirements_dev.txt # if not already installed
OR
$ make develop
```

Run quick tests (skip slow and online):

```
$ pytest -m 'not slow and not online'
```

Run all tests:

```
$ pytest
```

Check pep8:

```
$ flake8
```

1.3.3 Run tests the lazy way

Do the same as above using the Makefile.

```
$ make test
$ make test-all
$ make lint
```

1.3.4 Prepare a release

Update the Conda specification file to build identical `environments` on a specific OS.

Note: You should run this on your target OS, in our case Linux.

```
$ conda env create -f environment.yml
$ source activate c4cds
$ make clean
$ make install
$ conda list -n c4cds --explicit > spec-file.txt
```

1.3.5 Bump a new version

Make a new version of c4cds-wps in the following steps:

- Make sure everything is commit to GitHub.
- Update `CHANGES.rst` with the next version.
- Dry Run: `bumpversion --dry-run --verbose --new-version 0.8.1 patch`
- Do it: `bumpversion --new-version 0.8.1 patch`
- ... or: `bumpversion --new-version 0.9.0 minor`
- Push it: `git push`
- Push tag: `git push --tags`

See the [bumpversion](#) documentation for details.

1.4 Processes

- *CMIP5 Regridder*
- *CORDEX Subsetter*

1.4.1 CMIP5 Regridder

1.4.2 CORDEX Subsetter

1.5 Changes

1.5.1 0.2.0 (2019-11-01)

Added the following processes:

- CMIP5 regridder for the Copernicus C3S CMIP5 data archive.
- CORDEX subsetter (countries) for the Copernicus CORDEX data archive.

1.5.2 0.1.0 (2018-10-22)

- First release.

INDICES AND TABLES

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