name: "Testing and distribution"

on:

push:

pull\_request:

workflow\_dispatch:

jobs:

lint-python:

name: "Lint Python code"

runs-on: ubuntu-latest

steps:

- name: "Checkout"

uses: actions/checkout@v2

- id: install-pipenv

name: "Install pipenv"

run: pipx install pipenv

- id: install-python

name: "Install Python"

uses: actions/setup-python@v3

with:

python-version: '3.9'

cache: 'pipenv'

- id: install-python-dependencies

name: "Install Python dependencies"

run: pipenv install flake8

- id: lint

name: "Lint Python code"

run: pipenv run flake8 anss

test-python:

name: "Test Python"

runs-on: ubuntu-latest

services:

postgres:

image: postgis/postgis:latest

env:

POSTGRES\_USER: postgres

POSTGRES\_PASSWORD: postgres

POSTGRES\_DB: test

ports:

- 5432:5432

options: --health-cmd pg\_isready --health-interval 10s --health-timeout 5s --health-retries 5

strategy:

matrix:

python: ["3.8", "3.9", "3.10"]

steps:

- name: "Checkout the repo"

uses: actions/checkout@v3

- name: "psycopg2 prerequisites"

run: sudo apt-get install libpq-dev postgresql-client gdal-bin libgdal-dev

- name: "Install pipenv"

run: pipx install pipenv

- uses: actions/setup-python@v3

with:

python-version: '3.9'

cache: 'pipenv'

- id: pipenv-install

name: "Install Python dependencies"

run: pipenv install --dev --python `which python`

- name: "Test"

run: pipenv run python setup.py test

env:

PGPASSWORD: postgres

test-build:

name: "Build Python package"

runs-on: ubuntu-latest

needs: [test-python]

steps:

- name: "Checkout"

uses: actions/checkout@v3

- name: "Install pipenv"

run: pipx install pipenv

- uses: actions/setup-python@v3

with:

python-version: '3.9'

cache: 'pipenv'

- id: pipenv-install

name: "Install Python dependencies"

run: pipenv install --dev --python `which python`

- id: build

name: "Build release"

run:

pipenv run python setup.py sdist

pipenv run python setup.py bdist\_wheel

ls -l dist

- id: check

name: "Check release"

run:

pipenv run twine check dist/\*

- id: save

name: "Save artifact"

uses: actions/upload-artifact@v2

with:

name: "test-release-${{ github.run\_number }}"

path: ./dist

if-no-files-found: error

tag-release:

name: "Tagged PyPI release"

runs-on: ubuntu-latest

needs: [test-build]

if: github.event\_name == 'push' && startsWith(github.ref, 'refs/tags')

steps:

- uses: actions/setup-python@v2

with:

python-version: '3.9'

- id: fetch

name: "Fetch artifact"

uses: actions/download-artifact@v2

with:

name: "test-release-${{ github.run\_number }}"

path: ./dist

- id: publish

name: "Publish release"

uses: pypa/gh-action-pypi-publish@release/v1

with:

user: \_token\_

password: ${{ secrets.PYPI\_API\_TOKEN }}

verbose: true

verify\_metadata: false