

name: DARKBOT™ CI/CD Pipeline

on: push: branches: [ main, develop ] pull\_request: branches: [ main, develop ] schedule: - cron: '0 0 \* \* 0'  
# Weekly run on Sundays

jobs: lint: name: Code Quality runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v3

- name: Set up Python

uses: actions/setup-python@v4

with:

python-version: '3.10'

- name: Install dependencies

run: |

python -m pip install --upgrade pip

pip install flake8 black isort

if [ -f requirements.txt ]; then pip install -r requirements.txt; fi

- name: Lint with flake8

run: |

flake8 . --count --select=E9,F63,F7,F82 --show-source --statistics

flake8 . --count --exit-zero --max-complexity=10 --max-line-length=127 --statistics

- name: Check formatting with black

run: |

black --check darkbot tests

- name: Check imports with isort

run: |

isort --check-only --profile black darkbot tests

test: name: Test runs-on: ubuntu-latest strategy: matrix: python-version: [3.8, 3.9, '3.10']

steps:

- uses: actions/checkout@v3
- name: Set up Python `${{ matrix.python-version }}`  
uses: actions/setup-python@v4  
with:  
python-version: `${{ matrix.python-version }}`
- name: Install dependencies  
run: |  
python -m pip install --upgrade pip  
pip install pytest pytest-cov  
if [ -f requirements.txt ]; then pip install -r requirements.txt; fi  
pip install -e .
- name: Test with pytest  
run: |  
pytest tests/ --cov=darkbot --cov-report=xml
- name: Upload coverage to Codecov  
uses: codecov/codecov-action@v3  
with:  
file: ./coverage.xml  
fail\_ci\_if\_error: true

benchmark: name: Performance Benchmarks runs-on: ubuntu-latest needs: test if: github.event\_name == 'push' && github.ref == 'refs/heads/main'

steps:

- uses: actions/checkout@v3
- name: Set up Python
  - uses: actions/setup-python@v4
  - with:
    - python-version: '3.10'
- name: Install dependencies
  - run: |
    - python -m pip install --upgrade pip
    - pip install pandas matplotlib
    - if [ -f requirements.txt ]; then pip install -r requirements.txt; fi
    - pip install -e .
- name: Run benchmarks
  - run: |
    - python scripts/run\_benchmarks.py --iterations 10 --dimensions 128 --batch-size 8 --ci-mode
- name: Upload benchmark results
  - uses: actions/upload-artifact@v3
  - with:
    - name: benchmark-results
    - path: benchmark\_results/
- name: Update benchmark history
  - if: github.ref == 'refs/heads/main'
  - run: |
    - python scripts/update\_benchmark\_history.py

docs: name: Documentation runs-on: ubuntu-latest needs: test if: github.event\_name == 'push' && (github.ref == 'refs/heads/main' || github.ref == 'refs/heads/develop')

steps:

- uses: actions/checkout@v3
- name: Set up Python
  - uses: actions/setup-python@v4
  - with:
    - python-version: '3.10'
- name: Install dependencies
  - run: |
    - python -m pip install --upgrade pip
    - pip install sphinx sphinx\_rtd\_theme sphinxcontrib-napoleon
    - if [ -f requirements.txt ]; then pip install -r requirements.txt; fi
    - pip install -e .
- name: Build documentation
  - run: |
    - cd docs
    - make html
- name: Deploy documentation
  - if: github.ref == 'refs/heads/main'
  - uses: peaceiris/actions-gh-pages@v3
  - with:
    - github\_token: \${ secrets.GITHUB\_TOKEN }
    - publish\_dir: ./docs/\_build/html

gpu-test: name: GPU Tests runs-on: self-hosted if: github.ref == 'refs/heads/main' needs: test

steps:

- uses: actions/checkout@v3

- name: Set up Python

  - uses: actions/setup-python@v4

  - with:

    - python-version: '3.10'

- name: Install dependencies

  - run: |

    - python -m pip install --upgrade pip

    - if [ -f requirements.txt ]; then pip install -r requirements.txt; fi

    - pip install -e .

- name: Run GPU tests

  - run: |

    - python -m pytest tests/gpu\_tests/

- name: Run full-scale benchmarks

  - run: |

    - python scripts/run\_benchmarks.py --iterations 50 --dimensions 512 --batch-size 16 --

    - with-gpu

- name: Upload GPU benchmark results

  - uses: actions/upload-artifact@v3

  - with:

    - name: gpu-benchmark-results

    - path: benchmark\_results/gpu\_benchmarks/

release: name: Release Package runs-on: ubuntu-latest if: github.event\_name == 'push' &&  
startsWith(github.ref, 'refs/tags') needs: [test, docs, gpu-test]

steps:

- uses: actions/checkout@v3
  
- name: Set up Python  
uses: actions/setup-python@v4  
with:  
python-version: '3.10'
  
- name: Install dependencies  
run: |  
python -m pip install --upgrade pip  
pip install build twine
  
- name: Build package  
run: |  
python -m build
  
- name: Create GitHub Release  
uses: softprops/action-gh-release@v1  
with:  
files: |  
dist/\*.whl  
dist/\*.tar.gz
  
- name: Publish to PyPI  
env:  
TWINE\_USERNAME: \${ secrets.PYPI\_USERNAME }  
TWINE\_PASSWORD: \${ secrets.PYPI\_PASSWORD }  
run: |  
twine upload dist/\*