

## / Docker style

Docker style

```
- uses: ansys/actions/docker-style@main
  with:
    directory: docker
    error-level: 1
    recursive: true
```

*Ref:* [Docker style](#)

## / Release pypi public

Release to public pypi

```
- name: Release to the public PyPI repository
  uses: ansys/actions/release-pypi-public@main
  with:
    library-name: ansys-<product>-<library>
    twine-token: ${{ secrets.PYPI_TOKEN }}
    twine-username: __token__
```

*Ref:* [Release to public PyPI](#)

## / Doc deploy pr

Deploy pr documentation

```
- uses: ansys/actions/doc-deploy-pr@main
  with:
    bot-email: ${{ secrets.PYANSYS_CI_BOT_EMAIL }}
    bot-user: ${{ secrets.PYANSYS_CI_BOT_USERNAME }}
    cname: ${{ env.DOCUMENTATION_CNAME }}
    maximum-pr-doc-deployments: 3
    token: ${{ secrets.GITHUB_TOKEN }}
```

*Ref:* [Deploy PR documentation](#)

## / Hk package clean untagged

Perform untagged versions cleanup

```
- name: Perform untagged versions cleanup
  uses:
    ansys/actions/hk-package-clean-untagged@main
  with:
    package-name: mypackage
    token: ${{ secrets.GITHUB_TOKEN }}
```

*Ref:* [Perform untagged versions cleanup](#)

## / Build wheelhouse

Build wheelhouse for latest python versions

```
- name: Build a wheelhouse of the Python library
  uses: ansys/actions/build-wheelhouse@main
  with:
    library-name: <ansys-product-library>
```

```
operating-system: ${{ matrix.os }}
python-version: ${{ matrix.python-version }}
```

*Ref:* [Build wheelhouse for latest Python versions](#)

## / Doc deploy stable

Deploy stable documentation

```
- name: Deploy the stable documentation
  uses: ansys/actions/doc-deploy-stable@main
  with:
    bot-email: ${{ secrets.PYANSYS_CI_BOT_EMAIL }}
    bot-user: ${{ secrets.PYANSYS_CI_BOT_USERNAME }}
    cname: <library>.docs.pyansys.com
    token: ${{ secrets.GITHUB_TOKEN }}
```

*Ref:* [Deploy stable documentation](#)

## / Doc deploy changelog

Update changelog for new tag

```
- uses: ansys/actions/doc-deploy-changelog@main
  with:
    bot-email: ${{ secrets.PYANSYS_CI_BOT_EMAIL }}
    bot-user: ${{ secrets.PYANSYS_CI_BOT_USERNAME }}
    token: ${{ secrets.PYANSYS_CI_BOT_TOKEN }}
```

*Ref:* [Update CHANGELOG for new tag](#)

## / Tests pytest

Testing library with different operating systems and python versions

```
- name: Run pytest
  uses: ansys/actions/tests-pytest@main
  with:
    pytest-extra-args: --cov=ansys.<library>
    --cov-report=term
    --cov-report=html:.cov/html
    pytest-markers: -k 'mocked'
```

*Ref:* [Testing library with different operating systems and Python versions](#)

Optimized testing with different operating systems and python versions

```
- name: Login into the GitHub container registry
  uses:
    docker/login-action@f4ef78c080cd8ba55a85445d5b3
  with:
    password: ${{ secrets.GITHUB_TOKEN }}
    registry: ghcr.io
    username: ${{ secrets.GH_USERNAME }}
- env:
```

```
LICENSE_SERVER: ${{ secrets.LICENSE_SERVER }}
name: Pull, launch and verify desired Ansys
service
run: '...'
- name: Run pytest with desired markers and extra
arguments
uses: ansys/actions/tests-pytest@main
with:
  pytest-extra-args: ${{ matrix.cfg.extra-args }}
  pytest-markers: ${{ matrix.cfg.markers }}
  requires-xvfb: true
```

*Ref:* [Optimized testing with different operating systems and Python versions](#)

## / Doc style

Running documentation style checks

```
- name: Running documentation style checks
  uses: ansys/actions/doc-style@main
  with:
    token: ${{ secrets.GITHUB_TOKEN }}
```

*Ref:* [Running documentation style checks](#)

## / Check actions security

Check actions security

```
- uses: ansys/actions/check-actions-security@main
  with:
    auditing-level: high
    generate-summary: true
    token: ${{ secrets.GITHUB_TOKEN }}
    trust-ansys-actions: true
```

*Ref:* [Check actions security](#)

## / Build ci wheels

Build a c-extension library

```
- name: Build a C-extension library wheel artifacts
  uses: ansys/actions/build-ci-library@main
```

*Ref:* [Build a C-extension library](#)

## / Check licenses

Check library dependencies ship with valid licenses

```
- name: Validate third party licenses
  uses: ansys/actions/check-licenses@main
  with:
    python-version: ${{ env.MAIN_PYTHON_VERSION }}
```

*Ref:* [Check library dependencies ship with valid licenses](#)

## / Build library

### Build library basic example

```
- name: Build library source and wheel artifacts
  uses: ansys/actions/build-library@main
  with:
    library-name: ansys-<product>-<library>
```

*Ref:* [Build library basic example](#)

## / Code style

### Running code style checks

```
- name: Run PyAnsys code style checks
  uses: ansys/actions/code-style@main
```

*Ref:* [Running code style checks](#)

## / Doc changelog

### Create changelog fragment

```
- uses: ansys/actions/doc-changelog@main
  with:
    bot-email: ${{ secrets.PYANSYS_CI_BOT_EMAIL }}
    bot-user: ${{ secrets.PYANSYS_CI_BOT_USERNAME }}
    token: ${{ secrets.PYANSYS_CI_BOT_TOKEN }}
```

*Ref:* [Create changelog fragment](#)

## / Release pypi private

### Release to private pypi

```
- name: Release to the private PyPI repository
  uses: ansys/actions/release-pypi-private@main
  with:
    library-name: ansys-<product>-<library>
    twine-token: ${{ secrets.PYANSYS_PYPI_PRIVATE_PAT }}
    twine-username: __token__
```

*Ref:* [Release to private PyPI](#)

## / Hk automerger prs

### Automerging dependabot and pre-commit.ci prs

```
- name: Automerging dependabot and pre-commit.ci PRs
  uses: ansys/actions/hk-automerger-prs@main
  with:
    approver: ${{ secrets.PYANSYS_CI_BOT_USERNAME }}
    approver-token: ${{ secrets.PYANSYS_CI_BOT_TOKEN }}
```

*Ref:* [Automerging dependabot and pre-commit.ci PRs](#)

## / Release github

### Release to github

```
- name: Release to GitHub
  uses: ansys/actions/release-github@main
  with:
    library-name: ansys-<product>-<library>
```

*Ref:* [Release to GitHub](#)

## / Hk package clean except

### Perform versions cleanup - except certain tags

```
- name: Perform versions cleanup - except certain tags
  uses: ansys/actions/hk-package-clean-except@main
  with:
    package-name: mypackage
    tags-kept: latest, latest-unstable
    token: ${{ secrets.GITHUB_TOKEN }}
```

*Ref:* [Perform versions cleanup - except certain tags](#)

## / Check pr title

### Check pull-request title follows conventional commits

```
- uses: ansys/actions/check-pr-title@main
  with:
    token: ${{ secrets.GITHUB_TOKEN }}
```

*Ref:* [Check pull-request title follows conventional commits](#)

## / Doc deploy dev

### Deploy developers documentation

```
- name: Deploy the latest documentation
  uses: ansys/actions/doc-deploy-dev@main
  with:
    bot-email: ${{ secrets.PYANSYS_CI_BOT_EMAIL }}
    bot-user: ${{ secrets.PYANSYS_CI_BOT_USERNAME }}
    cname: <library>.docs.pyansys.com
    token: ${{ secrets.GITHUB_TOKEN }}
```

*Ref:* [Deploy developers documentation](#)

## / Doc build

### Installing additional system dependencies for building documentation

```
- name: Run Ansys documentation building action
  uses: ansys/actions/doc-build@main
  with:
    dependencies: graphviz mermaid-cli
```

*Ref:* [Installing additional system dependencies for building documentation](#)

### Building library documentation that using xvfb

```
- name: Run Ansys documentation building action
  uses: ansys/actions/doc-build@main
  with:
    requires-xvfb: true
```

*Ref:* [Building library documentation that using XVFB](#)

### Building library documentation

```
- name: Run Ansys documentation building action
  uses: ansys/actions/doc-build@main
```

*Ref:* [Building library documentation](#)

## / Check environment approval

### Check environment approval

```
- id: block
  name: Decide which environment should be used
  uses: ansys/actions/check-environment-approval@main
```

*Ref:* [Check environment approval](#)

## / Check vulnerabilities

### Check library vulnerabilities

```
- uses: ansys/actions/check-vulnerabilities@main
  with:
    dev-mode: ${{ github.ref != "refs/heads/main" }}
    python-package-name: ansys-<product>-<library>
    python-version: ${{ env.MAIN_PYTHON_VERSION }}
    token: ${{ secrets.PYANSYS_CI_BOT_TOKEN }}
```

*Ref:* [Check library vulnerabilities](#)

## / Release pypi test

### Release to test pypi

```
- name: Release to the test PyPI repository
  uses: ansys/actions/release-pypi-test@main
  with:
    library-name: ansys-<product>-<library>
    twine-token: ${{ secrets.PYANSYS_PYPI_TEST_PAT }}
    twine-username: __token__
```

*Ref:* [Release to test PyPI](#)