Application of CI/CD practices with GitHub actions

Herbert D. Ludowieg

Department of Chemistry, University at Buffalo, State University of New York

March 22, 2023

Outline

Outline for section 1

What is a CI/CD pipeline?

- CI: Continuous Integration
- CD: Continuous Delivery

What is a CI/CD pipeline?

- CI: Continuous Integration
- CD: Continuous Delivery
- A pipeline is a set of instructions usually to
  - Build code (CI): Compiling and installing depending on type of code
  - Test code (CI): Run a set of tests to ensure the code is working properly
  - Deployment (CD): Can be broken down into
    - Deployment of code and binaries
    - Deployment of documentation

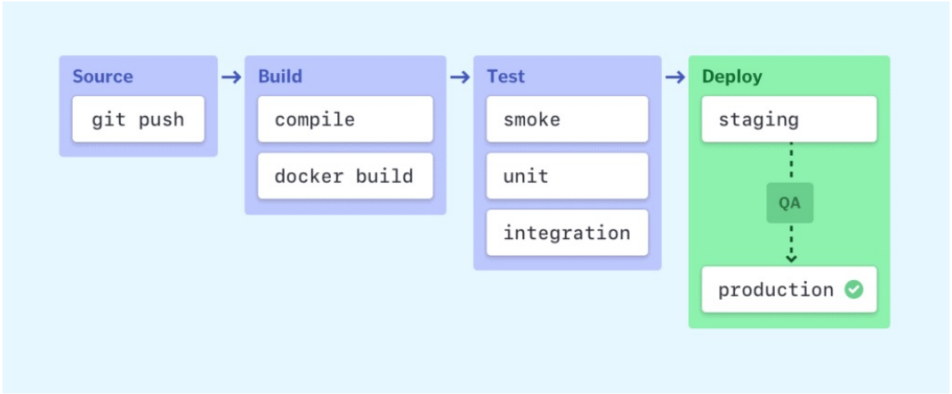
What is a CI/CD pipeline?

- CI: Continuous Integration
- CD: Continuous Delivery
- A pipeline is a set of instructions usually to
  - Build code (CI): Compiling and installing depending on type of code
  - Test code (CI): Run a set of tests to ensure the code is working properly
  - Deployment (CD): Can be broken down into
    - Deployment of code and binaries
    - Deployment of documentation
- This can be done locally, but it is not a good practice

What is a CI/CD pipeline?

- CI: Continuous Integration
- CD: Continuous Delivery
- A pipeline is a set of instructions usually to
  - Build code (CI): Compiling and installing depending on type of code
  - Test code (CI): Run a set of tests to ensure the code is working properly
  - Deployment (CD): Can be broken down into
    - Deployment of code and binaries
    - Deployment of documentation
- This can be done locally, but it is not a good practice
- Eliminate any human errors

Real-world application



Build stage

- Here we handle building/installing the code

Build stage

- Here we handle building/installing the code
- If we have a compiled language we include compilers
  - Do we include multiple compilers and versions?

Build stage

- Here we handle building/installing the code
- If we have a compiled language we include compilers
  - Do we include multiple compilers and versions?
    - Backwards compatibility

Build stage

- Here we handle building/installing the code
- If we have a compiled language we include compilers
  - Do we include multiple compilers and versions?
    - Backwards compatibility
  - What libraries are required when compiling?

Build stage

- Here we handle building/installing the code
- If we have a compiled language we include compilers
  - Do we include multiple compilers and versions?
    - Backwards compatibility
  - What libraries are required when compiling?
- If we have interpreted code we can skip this

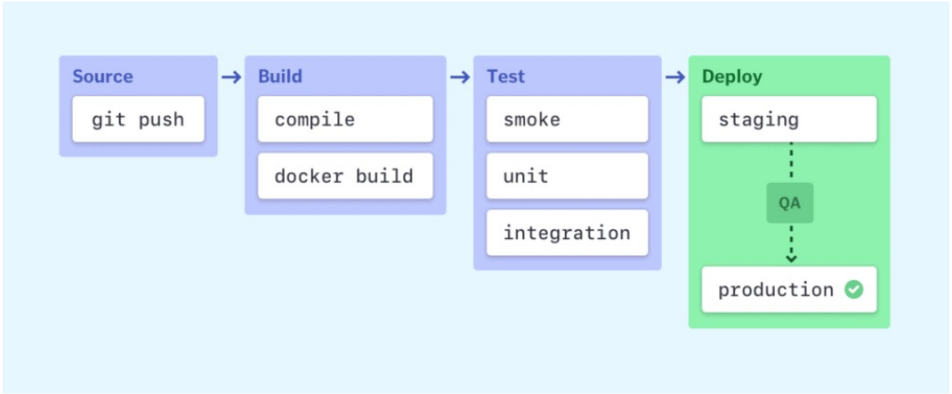
Build stage

- Here we handle building/installing the code
- If we have a compiled language we include compilers
  - Do we include multiple compilers and versions?
    - Backwards compatibility
  - What libraries are required when compiling?
- If we have interpreted code we can skip this
- Backwards compatibility: The ability of a system, hardware or software, to successfully interface with older versions

Build stage

- Here we handle building/installing the code
- If we have a compiled language we include compilers
  - Do we include multiple compilers and versions?
    - Backwards compatibility
  - What libraries are required when compiling?
- If we have interpreted code we can skip this
- Backwards compatibility: The ability of a system, hardware or software, to successfully interface with older versions
- Docker build: TL;DR create a new blank environment to build everything. Mimic a new user using a program for the first time.

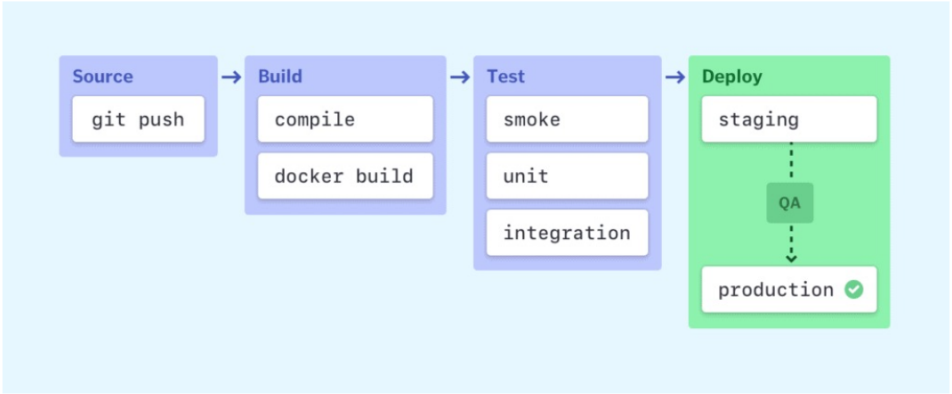
Real-world application



Test stage

- Now we want to run automated test cases to validate code's correctness and behavior
  - This is the safety net of every program
  - Ensures that the new code is not introducing any obvious bugs
  - Can be a good check for backwards compatibility
- This is an important step of every developer
- Should be second nature to write code and tests in conjunction
- There can be different levels to tests
  - Smoke tests
  - Unit tests
  - Integration tests
  - End-to-end tests
  - Load testing

Real-world application

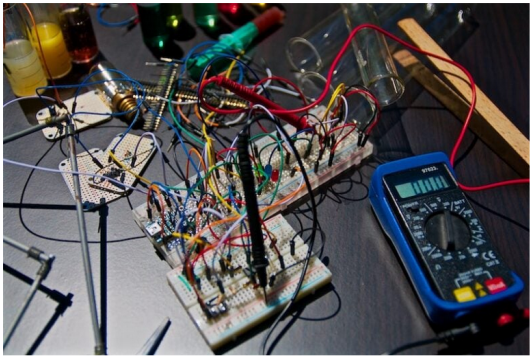


Deployment stage

- Here we take care of the final stages of the pipeline and begin the process of releasing the code.
- Begin by creating a staging environment where code can be reviewed
  - Typically something like GitHub where a developer release exists
- Perform code quality checks and code coverage
  - Codacy for many Python projects
- Can also deploy documentation for developer version

Outline for section 2

Smoke testing



- Simple tests that will catch any immediate errors or bugs
- Does the circuit go up in flames when we connect the power

Unit tests

- Unit testing: Test a specific unit of code
- Can be considered the most important part of coding
- Should be a minimal working example of the code

Unit tests

- Unit testing: Test a specific unit of code
- Can be considered the most important part of coding
- Should be a minimal working example of the code
  - Should be an aid in what the code should be doing

Unit tests

- Unit testing: Test a specific unit of code
- Can be considered the most important part of coding
- Should be a minimal working example of the code
  - Should be an aid in what the code should be doing
  - Should be quick

Unit tests

- Unit testing: Test a specific unit of code
- Can be considered the most important part of coding
- Should be a minimal working example of the code
  - Should be an aid in what the code should be doing
  - Should be quick
- This can be the only way to make sure that the code is behaving properly
- Should be second nature to any programmer

```
import numpy as np
def get_triu(arr, k=0):
    if isinstance(arr, (list, tuple)):
        arr = np.array(arr)
    if arr.shape[0] != arr.shape[1]:
        raise ValueError("The input matrix must be square to get " \
                         +"the upper triangular elements")
    triu = np.triu_indices_from(arr, k=k)
    triu_arr = arr[triu]
    return triu_arr
```

- Code is supposed to return the upper triangular elements of a matrix

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 3 & 5 & 6 & 9 \end{pmatrix}$$

```
from math_funcs import get_triu
x = [[1,2,3],[4,5,6],[7,8,9]]
get_triu(x)
```

- What if we give the code a non-square matrix?

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 3 & 5 & 6 \end{pmatrix}$$

```
import numpy as np
def get_triu(arr, k=0):
    if isinstance(arr, (list, tuple)):
        arr = np.array(arr)
    if arr.shape[0] != arr.shape[1]:
        raise ValueError("The input matrix must be square to get " \
                         +"the upper triangular elements")
    triu = np.triu_indices_from(arr, k=k)
    triu_arr = arr[triu]
    return triu_arr
```

- What if we give the code a non-square matrix?

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 3 & 5 & 6 \end{pmatrix}$$

```
from math_funcs import get_triu
x = [[1,2,3],[4,5,6]]
get_triu(x)
```

Unit testing script with using pytest

```
mat_22 = [[1,2],[3,4]]
mat_33 = [[1,2,3],[4,5,6],[7,8,9]]
mat_23 = [[1,2,3],[4,5,6]]
@pytest.mark.parametrize('arr,k,actual',
                                    [(mat_22, 0, [1,2,4]),
                                     (mat_33, 0, [1,2,3,5,6,9]),
                                     (mat_23, 0, None)])
def test_get_triu(arr, k, actual):
    if actual is not None:
        triu_arr = get_triu(arr, k)
        assert np.allclose(triu_arr, actual)
    else:
        with pytest.raises(ValueError):
            _ = get_triu(arr, k)
```

Integration tests

- Usually this is testing how a module is working
- Test how the different components of the code base work together

Integration tests

- Usually this is testing how a module is working
- Test how the different components of the code base work together
- Are the different units of code passing the data correctly between each other?

Integration tests

- Usually this is testing how a module is working
- Test how the different components of the code base work together
- Are the different units of code passing the data correctly between each other?
- Usually much more intensive testing than unit testing
- Lays the foundation for higher level of tests like end-to-end testing

End-to-end (E2E) and load testing

- End-to-end tests
  - Usually a full run of the entire code-base
  - Essentially, it tests the user experience
  - Similar to a manual test
  - More applicable when a UI or digitized results are utilized

End-to-end (E2E) and load testing

- End-to-end tests
  - Usually a full run of the entire code-base
  - Essentially, it tests the user experience
  - Similar to a manual test
  - More applicable when a UI or digitized results are utilized
- Load testing
  - Run a large scale test on code
  - Check stability of code
  - Are there any memory leaks?
  - Is the code exceeding any parameters?

Outline for section 3

GitHub actions

- A CI/CD platform that can be used to run the entire pipeline
- Can build code, test it, generate documentation, and deploy
- We can do this for any pull request and push to a branch
- Automates the build steps as it creates a docker environment based on the given configuration
  - Can define multiple OS's to use including Ubuntu (Linux), MacOS, and Windows
  - Can specify which program versions to use
  - Can even specify which combinations you don't want to include

Python Example

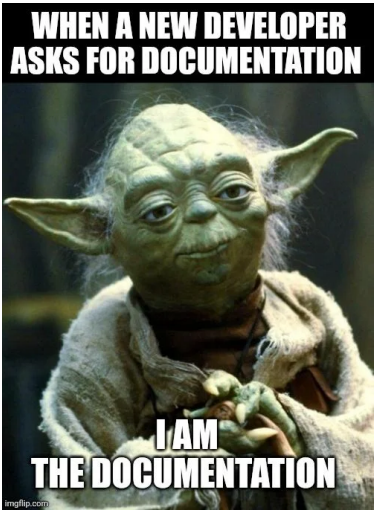
Will switch to a terminal window with a script from my own VibrAv package.

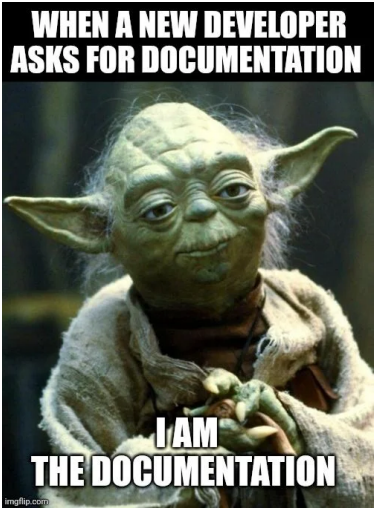
FORTRAN Example

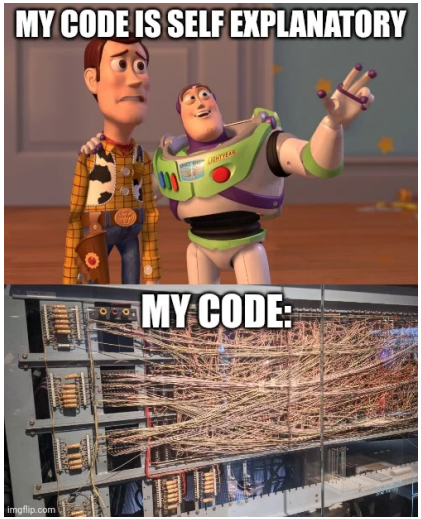
Will switch to a terminal window with the script used for the MCD-molcas repo.

Documentation

Documentation


WHEN A NEW DEVELOPER ASKS FOR DOCUMENTATION
I AM THE DOCUMENTATION
imgflip.com


WHEN A NEW DEVELOPER ASKS FOR DOCUMENTATION
I AM THE DOCUMENTATION
imgflip.com


MY CODE IS SELF EXPLANATORY
MY CODE:
imgflip.com

Documentation

- Documentation is often overlooked, but it can be more important than the actual code

Documentation

- Documentation is often overlooked, but it can be more important than the actual code
  - What will happen once you're done with the code?

Documentation

- Documentation is often overlooked, but it can be more important than the actual code
  - What will happen once you're done with the code?
  - How will the next person take over the code?

Documentation

- Documentation is often overlooked, but it can be more important than the actual code
  - What will happen once you're done with the code?
  - How will the next person take over the code?
- Documentation is one of the first things to be used to understand code
- Good documentation increases the lifespan of a code

Documentation

- Documentation is often overlooked, but it can be more important than the actual code
  - What will happen once you're done with the code?
  - How will the next person take over the code?
- Documentation is one of the first things to be used to understand code
- Good documentation increases the lifespan of a code
- GitHub has a service with GitHub pages where you can publish HTML pages
- This can be done every time a workflow with GitHub actions is triggered
  - Recommended to only be triggered when the master branch is updated

Writing documentation pages

- There are many ways that one can write documentation for code
- Most people think that if you write documentation pages you have to manually write them

Writing documentation pages

- There are many ways that one can write documentation for code
- Most people think that if you write documentation pages you have to manually write them
- **No!**

Writing documentation pages

- There are many ways that one can write documentation for code
- Most people think that if you write documentation pages you have to manually write them
- **No!**
- Should already be writing them in the source code
- There exist many third-party programs that can extract the docstrings in the source code
  - Sphinx: widely used for Python
  - FORD: can be used for FORTRAN

Sphinx

- It can automatically build all the documentation pages from the Python docstrings in the source code
- Create production level documentation pages in a variety of formats, like HTML and PDF with LaTeX
- Insert a Jupyter notebook into the documentation for tutorials
- Insert mathematical functions with very good support for LaTeX math functions
- Vibronic functions docs

Figure: Sphinx logo

GitHub Pages

- A service by GitHub to host documentation for open-source projects
- Can publish HTML pages on a website like https://herbertludowieg.github.io/vibrav
- Can even create a custom hostname for the project (not sure if this is free)
- Can actually use this as a portfolio as well as long as you design the pages yourself

Figure: GitHub Pages logo

Code quality checks

- Coding is just another form of writing and it is up to the author to choose how well it is written

Figure: Codacy logo

Code quality checks

- Coding is just another form of writing and it is up to the author to choose how well it is written
  - How easy is is to read?
  - Are there any vulnerabilities?

Figure: Codacy logo

Code quality checks

- Coding is just another form of writing and it is up to the author to choose how well it is written
  - How easy is is to read?
  - Are there any vulnerabilities?
  - Are any built-in functions being overwritten? (Python rant)

Figure: Codacy logo

Code quality checks

- Coding is just another form of writing and it is up to the author to choose how well it is written
  - How easy is is to read?
  - Are there any vulnerabilities?
  - Are any built-in functions being overwritten? (Python rant)
- Another metric is code coverage, or how much of the code is tested by unit tests
  - The higher this metric is the easier it can be to detect bugs
  - 100% coverage is not realistic
  - It is quite relative to the code base

Figure: Codacy logo

Code quality checks

- There are many third-party programs that can do this
  - Codacy
  - Coveralls
- These are just tools to assist in the reviewing process
- Important that maintainers do a manual check
- Programs can highlight some things as issues when they may not actually be an issue
- Important tool to increase readability of code base
- vibronic.py duplication
- VibrAv main page
- vibronic func coverage

Figure: Codacy logo

Outline for section 4

Summary

- When writing any code it is important to follow the steps in the CI/CD pipeline
- Increases the lifespan of code base and makes it easier for others to develop the code
- Help to maintain a consistent quality for the code
- Important to create guidelines for contributing to the code
- Hope this serves as a good initial exposure to what is out there in terms of tools for your coding projects
- Can be applied to any aspect when dealing with code
- **Always document what you are doing**


THIS CODE...
WRITTEN LAST MONTH
...I HAVE
NO MEMORY
OF THIS PLACE
imgflip.com