Continuous integration and deployment for NMRLipids

By Magnus Sletten

Contents of presentation

What is accomplished and some future plans

Github-Actions: best practices and practical examples

Docker: how to integrate docker images with Github Actions

NREC: How to utilise NREC resources for the project and some improvements that can be made here

The current status

What is currently done:

- Documentation is built and deployed automatically
- Automatic run of unit tests using NREC

What needs some work until deployment:

- Automatic processing of new simulations via website.

Github Actions

Repository wide event system keeps track of when something happens

Events can trigger automated workflows which we can define

An github workflow: description of what happens when event is triggered

Pre existing actions/ custom actions

Let’s see an example of a Github workflow

Example Workflow

```yaml
name: Print and Run Advanced Script

on:
  push:
    branches: [ "main" ]

jobs:
  print_message:
    runs-on: ubuntu-latest
    steps:
      - name: Run a one-line script
        run: echo Hello, world!

  run_advanced_script:
    needs: print_message
    runs-on: ubuntu-latest
    steps:
      - name: Run advanced molecular simulation
        run: |
          number_of_atoms=$((1 + 1))
          echo "Number of atoms: $number_of_atoms"
```

Some simple tips:

Each job runs in a fresh environment.

For sequential tasks:

1: Group directly connected tasks into one job.

2: Pass data between jobs via artifacts.

Smaller data amounts can be saved as outputs.

Let's improve the workflow we made

Goal: Let's see if we can improve the structure of the workflow

Modular reusable code is good! (sometimes)

Splitting up the workflow in smaller workflow files using trigger: workflow_call

These smaller workflows can be used as "methods" in other workflows

Making a reusable printing workflow

```yaml
# An action to print a message, which can be used in other
workflows.
name: Print Message

on:
  workflow_call:
    inputs:
      message:
        type: string
        description: "Message you want to print:"
        default: "hello world"

jobs:
  Print_Message:
    runs-on: ubuntu-latest
    steps:
      - name: Run a one-line script
        run: echo ${{inputs.message}}
```

Making a reusable molecular simulation workflow

```yaml
name: Run Molecular Simulation Script

on:
  workflow_call:
    inputs:
      number_of_hydrogen:
        type: string
        description: "Number of hydrogen atoms"
        default: "1"
      number_of_oxygen:
        type: string
        description: "Number of oxygen atoms"
        default: "1"
jobs:
  run_advanced_script:
    runs-on: ubuntu-latest
    steps:
      - name: Run advanced molecular simulation
        run: |
          number_of_hydrogen${{ inputs.number_of_hydrogen }}
          number_of_oxygen${{ inputs.number_of_oxygen }}
          total_atoms=$(( number_of_hydrogen + number_of_oxygen ))
          echo "Number of atoms: $total_atoms"
```

Referencing them in main-action

```yaml
name: Improved print and script execution
on:
  push:
    branches: [ "main" ]

jobs:
  print_message:
    uses: ./.github/workflows/print_message.yml
    with:
        message: "This action looks better now!"

  run_simulation:
    needs: print_message
    uses: ./.github/workflows/run_simulation.yml
    with:
        number_of_hydrogen: "2"
        number_of_oxygen: "2"
```

Github Actions and repository forks

Trigger: “On: Pull-request”

Workflows from forks are used

Token without write permission

No access to repo secrets

Can require permission from admin

Trigger “On: Pull-request-target”

Workflows from main repository are used

Token with write permission

Has access to secrets

When code is run from the forks: additional security should be added.

Example of safe workflow using code from forks:

```yaml
# Running hello script from fork, but with better security:
name: Run Hello Script
permissions:
  contents: read
on:
  pull_request_target:
    branches: "main"

jobs:
    Run_Pull_Request_Code:
        runs-on: ubuntu-latest
        steps:
          - name: Checkout pull-request code
            uses: actions/checkout@v4
            with:
                repository: ${{ github.event.pull_request.head.repo.full_name }}
                ref:        ${{ github.event.pull_request.head.sha }}
                fetch-depth: 1
          - name: Run hello script
            run: ./print_hello.sh
```

Adding repository name check:

Actions being run accidentally in forks

Very simple to fix:

```yaml
jobs:
  Run-tests:
    if: github.repository == 'NMRLipids/Databank'
```

Docker: containerization

Creating consistent environments

Independent of the current operating system

Dependencies pre installed like Gromacs

Perfect for integration with CI/CD like Github Actions

Process:

- 1: Creating a docker file which describes the environment
- 2: Building a docker image from the description
- 3: Pushing the image to Dockerhub

Integrating Docker with Github Actions

Dockerhub images are easily integrated

Example:

```
    build-documentation:
        if: github.repository == 'NMRLipids/Databank'
        runs-on: ubuntu-latest
        container:
           image: nmrlipids/doc-builder
```

Using Runner Image on NREC

NREC: Cloud compute platform: supplies resources we can use for compute

Connecting Github repository and NREC resources: Github runner.

Current state could be improved!

Initially in testing: automate registration with repository token.

Security concerns with token requirements.

Possible solution: registering runner with organisation.

Then: Fine grained access token which can register runners.

This also allows other repositories to utilise the runners

NREC

So far we have seen default github runners being used.

This is perfect for smaller automated tasks.

For larger tasks: *use our registered runners*

Just change the reference after “runs-on”, here it’s changed to nrec-large

“Nrec-large” is a tag for a runner.

There can be multiple runners with this tag.

```
Tests:
  if: github.repository == NMRlipids/Databank'
  runs-on: nrec-large
  container:
    image: nmrlipids/core
```