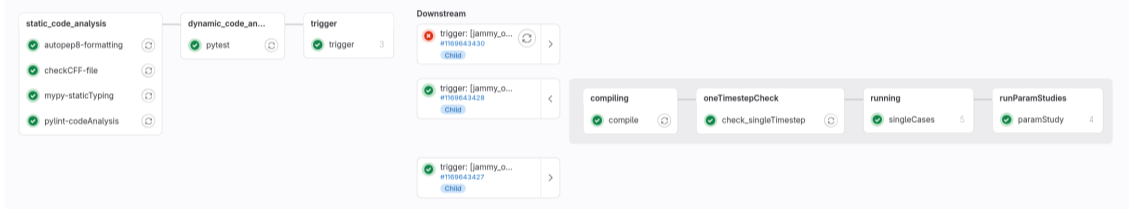


Applied CI: an academic OpenFOAM example



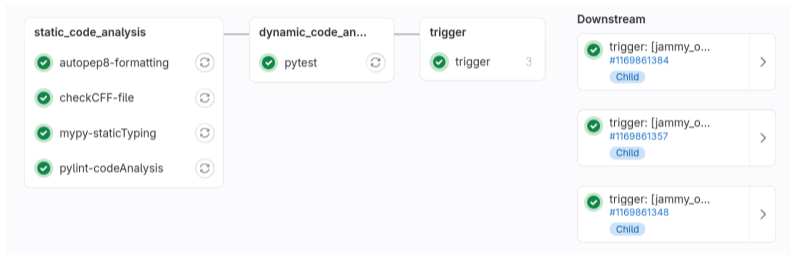
TECHNISCHE
UNIVERSITÄT
DARMSTADT

Engineering Research Software - NFDI4Ing Community meets Archetype BETTY 2024



Upstream pipeline

Sanity checks and container building



- In the first stage the formal status of the repository is checked, e.g. formatting.
- In the second stage *pytest* runs on python scripts we need for evaluation, i.e. of parameter studies.
- In the optional third stage, a docker image with the needed packages is build. This takes ~2.5h and is only done, once a new version of OpenFOAM is available.
- The fourth(/third) stage triggers the *downstream* OpenFOAM pipeline.

Upstream pipeline

The gitlab-ci.yml-file



- Our `.gitlab-ci.yml`-file only has 78 lines.
- For brevity we include jobs from other files.
- Three variables are used to steer the pipeline.
- Rules also steer the pipeline execution.

```
1 stages:
2   - build_and_analyze
3   - transfer_and_analyze
4   - build
5
6 include:
7   - ./jobs_build_and_analyze_pipeline.yml
8   - ./jobs_build_and_analyze_pipeline.yml
9   - ./jobs_build_and_analyze_pipeline.yml
10
11 variables:
12   PIPELINE_DEFINITION: "upstream"
13   PIPELINE_NAME: "name"
14   PIPELINE_ID: "ID"
15   PIPELINE_DESCRIPTION: "description"
16   PIPELINE_NAMESPACE: "namespace"
17
18 include_job_template:
19
20   include:
21     - CI_PROJECT_DIR: "${CI_PROJECT_DIR}/${CI_PROJECT_NAME}/${CI_PROJECT_NAMESPACE}"
22     # deprecated usage: "name.namespace"
23
24   if: always
25   # variables: name.namespace
26   # if: always
27   # if: never
28   # if: failure
29   # if: success
30   # if: skipped
31
32   when: always
33
34   rules:
35     # avoid duplicate pipelines, in case merge request is open
36     - if: $CI_PIPELINE_SOURCE == "merge_request_event"
37       when: never
38     # disable the CI by starting the commit via "top" or "skip" "next"
39     - if: $CI_COMMIT_TITLE == "top" || $CI_COMMIT_TITLE == "skip"
40       when: never
41     - when: always
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
59
60 .rule:
61 rules:
62   ## run always, if on master branch
63   - if: $PARENT_PIPELINE_COMMIT_BRANCH == $PARENT_PIPELINE_DEFAULT_BRANCH
64   ## run always, if on dev branch
65   - if: $PARENT_PIPELINE_COMMIT_BRANCH == "dev"
66   ## run always for selected (current) version
67   - if: $IMAGETAG == $CI_IMAGE
68   ## else: can start manual
69   - when: manual
70
```

- Rules can be applied to the whole pipeline or to individual jobs.

The docker build job

The gitlab-ci.yml-file



The `docker_build` job syntax:

```
42
43 docker_build:
44   stage: build
45   image: docker:20.10.16-dind
46   services:
47     - docker:20.10.16-dind
48   before_script: []
49   rules:
50     - if: $BUILDIMAGE == "true"
51   script:
52     - echo $IMAGETAG
53     - echo $CI_REGISTRY_PASSWORD | docker login $CI_REGISTRY -u $CI_REGISTRY_USER --password-stdin
54     - >
55       docker build
56       -t $CI_REGISTRY_IMAGE:$IMAGETAG
57       -t $CI_REGISTRY_IMAGE:latest
58       --build-arg UBUNTUVERSIONARG=${UBUNTUVERSION} --build-arg OPENFOAMVERSIONARG=$OPENFOAMVERSION .
59     - ls
60     - docker push --all-tags $CI_REGISTRY_IMAGE
61
62
63 trigger:
64   stage: trigger
65   allow_failure: true
66   needs:
67     # if docker_build job is created, it is needed
68     - job: docker_build
69     optional: true
70     - job: pytest
```

Remember: This takes ~2.5h and is only done, once a new version of OpenFOAM is available.

- The pipeline "spins" a docker container from a docker-in-docker image.
- Inside this docker environment, we build a docker image from the docker-file inside the repo.
 - ▣ Install dependencies like e.g. numpy, pandas and Jupyter Notebooks.
 - ▣ Install a specific OpenFOAM-version.
 - ▣ At this point, don't compile "our" code. We use this image as a base image.
- We push this docker image to the GitLab *container registry*.
- When successful the pipeline continues (see *trigger-job*, keyword *needs*).

The trigger job

The gitlab-ci.yml-file



The *matrix* definition:

```
18
19 .matrix_definition:
20   parallel:
21     matrix:
22       - CI_IMAGE: ["jammy_openfoam2212", "jammy_openfoam2306", "jammy_openfoam2312"]
23         # deprecated images: "jammy_openfoam2206"
24     ## Usage
25     # extends: .matrix_definition
26     ## or
27     # parallel:
28     #   !reference [.matrix_definition, parallel]
29
```

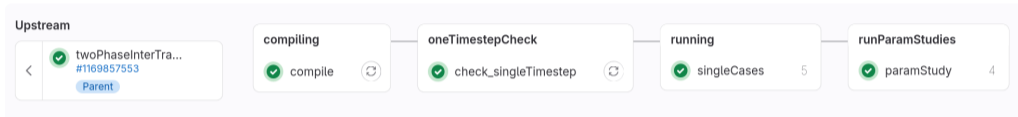
The *trigger* job syntax:

```
62
63 trigger:
64   stage: trigger
65   allow_failure: true
66   needs:
67     # if docker_build job is created, it is needed
68     - job: docker_build
69     optional: true
70     - job: pytest
71   extends: .matrix_definition
72   trigger:
73     include:
74       - local: .child.yml
75   variables:
76     PARENT_PIPELINE_COMMIT_BRANCH: $CI_COMMIT_BRANCH
77     PARENT_PIPELINE_DEFAULT_BRANCH: $CI_DEFAULT_BRANCH
78
```

- Run if *pytest* and *docker_build* were successful.
- Trigger (=create) new pipeline from *.child.yml*.
- Pass variables `$CI_COMMIT_BRANCH` and `$CI_DEFAULT_BRANCH` to downstream pipeline.
- Use *matrix* to create the downstream pipeline for multiple OpenFOAM versions.
 - ▣ Why downstream-pipelines?
Matrix-downstream pipelines might be the simplest way to pass compile-artifacts of OpenFOAM between jobs, when testing multiple OpenFOAM versions.

Downstream pipeline

OpenFOAM project compiling and testing



- Build our project inside our docker container with the standard OpenFOAM and dependencies.
- Run one timestep of all simulations and check for any errors and warnings during this.
 - ▣ We search the log files for errors or warnings.
 - ▣ Find warnings, you have never seen!
- Run simulations.
 - ▣ Check if tests run before running more time-consuming parameter studies.
- Run parameter studies.
 - ▣ Strictest form of (our) testing: convergence testing.

DRY: Don't Repeat Yourself

Defining default and snippets: compiling and sourcing OpenFOAM



```
7
8 default:
9 image:
10 name: $CI_REGISTRY_IMAGE:$CI_IMAGE
11 before_script:
12 - !reference [ .source_openfoam, script ]
13
14 ## Re-use user-defined OF applications from the FOAM_USER_APPBIN artifact
15 - if [ ! -d "$FOAM_USER_APPBIN" ]; then mkdir -p "$FOAM_USER_APPBIN"; fi
16 - cp -r FOAM_USER_APPBIN/* $FOAM_USER_APPBIN
17 ## Re-use user-defined OF libraries from the FOAM_USER_LIBBIN artifact
18 - if [ ! -d "$FOAM_USER_LIBBIN" ]; then mkdir -p "$FOAM_USER_LIBBIN"; fi
19 - cp -r FOAM_USER_LIBBIN/* $FOAM_USER_LIBBIN
20
21 - ./Allwmake
22 - cd src
23 - wmake libso
24 - cd ..
25
26 - pip install seaborn
27
28 .source_openfoam:
29 script:
30 - ls /usr/lib/openfoam/
31 - OF_VERSION=$(ls /usr/lib/openfoam/)
32 - echo $OF_VERSION
33 - source /usr/lib/openfoam/$OF_VERSION/etc/bashrc || true
34 - echo $WM_PROJECT_DIR
35
```

All jobs executing OpenFOAM need to

- use our self-build image,
- source OpenFOAM and
- handle OpenFOAM compilation artifacts.

Exploit that!

For repetitive tasks, use features like:

- *default* is executed, if not specified otherwise.
- *before_script* is executed before the "normal" script.
- Snippets can be defined and called from within *script* or *before_script*.

Allow failure: our compile job

The gitlab-ci.yml-file



```
70
71 compile:
72   stage: compiling
73   rules:
74     - !reference [.rule, rules]
75   allow_failure:
76     exit_codes: 42
77   before_script: []
78   script:
79     - !reference [.source_openfoam, script]
80
81   ## Allmake project and log output
82   # - ./Allmake > log.Allmake 2>&1
83   # - cat log.Allmake
84   - ./Allmake 2>&1 | tee log.Allmake
85
86   ### Checking for compilation/Linker errors that don't abort usually
87   - COMPILATIONERRORS=0
88   - COMPILATIONERRORS=$(grep -ice "error" log.Allmake 2>&1) || true
89   - COMPILATIONWARNINGS=0
90   - COMPILATIONWARNINGS=$(grep -ice "warning" log.Allmake 2>&1) || true
123   - mkdir FOAM_USER_LIBBIN && cp -r $FOAM_USER_LIBBIN FOAM_USER_LIBBIN
124   - mkdir FOAM_USER_APPBIN && cp -r $FOAM_USER_APPBIN/* FOAM_USER_APPBIN
125
126
127   ## If there were errors exit with error code
128   - if [ $COMPILATIONERRORS -ne 0 ]; then exit 1; fi
129   - if [ $LIBERRORS -ne 0 ]; then exit 1; fi
130
131   ## If there were warnings exit with error code that is tolerated
132   - if [ $COMPILATIONWARNINGS -ne 0 ]; then exit 42; fi
133   - if [ $LIBWARNINGS -ne 0 ]; then exit 42; fi
134   - echo "No errors or warnings found!"
135
136 artifacts:
137   when: always
138   paths:
139     - log.*
140     - src/*
141     - solvers/*
142     - utilities/*
143     - FOAM_USER_LIBBIN/*
144     - FOAM_USER_APPBIN/*
```



- The job automatically fails for the worst errors.
- If there are errors in the job logs, then fail too.
- If there are warnings in the job logs (e.g. *unused variables*), then show a warning but continue the pipeline.

One-timestep-testing

Find errors fast



```
346
347 check_singleTimestep:
348   stage: oneTimestepCheck
349   rules:
350     - reference [.rvls, rvls]
351   script:
352     - python3 --version || true
353     - python3 tutorials_test_all.py
354
355   - echo "Done!"
356
357 artifacts:
358   when: always
359   paths:
360     - log.*
361     - tutorials_alltest/**/log.*
362
```

- Automatically find all simulation cases.
- Automatically run one timestep. Common steps:
 - ▣ cleaning of the case,
 - ▣ meshing,
 - ▣ (compiling of case specific utilities,)
 - ▣ execution of solver.
- Automatically search all log files for errors and warnings.

At the moment four cases are ignored, because of external dependencies.

```
924 Failed on job execution (3 cases):
925 ./tutorials_alltest/speciesTransfer10test/singlecase
926 ./tutorials_alltest/droplets/migratingParangon1droplet2D/singlecase
927 ./tutorials_alltest/droplets/expandingContaminatedDroplet1d
928 Error/warning found in logs (9 cases):
929 ./tutorials_alltest/deformation3D
930 ./tutorials_alltest/droplets/0droplet2D
931 ./tutorials_alltest/droplets/migratingParangon1droplet2D/singlecase
932 ./tutorials_alltest/droplets/rotatingContaminatedDroplet2D/singlecase
933 ./tutorials_alltest/droplets/expandingContaminatedDroplet1d
934 ./tutorials_alltest/droplets/expandingContaminatedDropletAxisymmetric/singlecase
935 ./tutorials_alltest/droplets/contaminatedDroplet2D
936 ./tutorials_alltest/shear2D
937 ./tutorials_alltest/droplets/meshing3D
938 Missing solver log (2 cases):
939 ./tutorials_alltest/deformation3D/log_pimpleFoam
940 ./tutorials_alltest/shear2D/log_pimpleFoam
941 Parameter studies (3 cases):
942 WMDNS: skipped execution
943 ./tutorials_alltest/igwWedgeStudy
944 ./tutorials_alltest/speciesTransfer10test
945 ./tutorials_alltest/droplets/migratingParangon1droplet2D
946 ./tutorials_alltest/droplets/rotatingContaminatedDroplet2D
947 ./tutorials_alltest/droplets/expandingContaminatedDropletAxisymmetric
948 Ignored cases (4 cases):
949 WMDNS: skipped execution
950 ./tutorials_alltest/implicitSurfaceTension1d/shooting2D/Allrun
951 ./tutorials_alltest/implicitSurfaceTension1d/droplet2D/Allrun
952 ./tutorials_alltest/shearCentrifugal2D/Allrun
953 ./tutorials_alltest/en11sphereTest/Allrun
954 No problem found:
955 ./tutorials_alltest/shooting2D
956 ./tutorials_alltest/capillaryRiseCases/capillaryRiseAxisymmetric1d
957 ./tutorials_alltest/capillaryRiseCases/capillaryRise2D
958 ./tutorials_alltest/capillaryRiseCases/capillaryRiseAxisymmetric_rise
959 ./tutorials_alltest/capillaryRiseCases/capillaryRiseAxisymmetric
960 ./tutorials_alltest/bubbles/bubble2D
961 ./tutorials_alltest/bubbles/bubble2D
962 ./tutorials_alltest/bubbles/twoPhaseContaminatedBubble2D
963 ./tutorials_alltest/bubbles/twoPhaseBubble2D
964 ./tutorials_alltest/bubbles/contaminatedBubble2D
965 ./tutorials_alltest/bubbles/twoPhaseContaminatedBubbleAxisymmetric
966 ./tutorials_alltest/bubbles/twoPhaseBubbleAxisymmetric
967 ./tutorials_alltest/bubbles/singlePhaseBubble3D_poly
968 ./tutorials_alltest/bubbles/twoPhaseBubble2D
969 ./tutorials_alltest/contactAngleSteady
970 ./tutorials_alltest/spm/droplets/singlecase
971 ./tutorials_alltest/capillaryRise3D
972 ./tutorials_alltest/droplets/twoPhaseDroplet2D
973 ./tutorials_alltest/droplets/twoPhaseDropletAxisymmetric
974 Ran 29 jobs, out of which 19 seem working.
975 Exiting with error
```



```
163
164 singleCases:
165   stage: running
166   rules:
167     !reference [.rule, rules]
168   parallel:
169     matrix:
170       - TUTORIAL_PATH:
171         - "speciesTransfer1Dtest/singlecase"
172         - "sgsWedgeStudy/singlecase"
173         - "droplets/rotatingContaminatedDroplet2D/singlecase"
174         - "droplets/expandingContaminatedDroplet3dTp1"
175         - "droplets/expandingContaminatedDropletAxisymmetric/singlecase"
176     script:
177       - echo "$TUTORIAL_PATH"
178       - cd tutorials/${TUTORIAL_PATH}
179       - !reference [.check_required_of_version, script]
180       - ./Allrun
181       - ls
182
183     - jupyter nbconvert --to notebook --inplace --execute plotOverTime.ipynb
184     - jupyter nbconvert plotOverTime.ipynb --execute --to markdown
185     - jupyter nbconvert plotOverTime.ipynb --execute --to html #pdf needs xelatex
186
187     - if [ "$(cat result.txt)" = "true" ]; then echo "The content is true!"; fi
188     - if [ "$(cat result.txt)" != "true" ]; then exit 1; fi
189
190 artifacts:
191   when: always
192   paths:
193     - tutorials/${TUTORIAL_PATH}/log.*
194     - tutorials/${TUTORIAL_PATH}/plot*
195     - tutorials/${TUTORIAL_PATH}/result.txt
196
```

We again use the *matrix*-keyword -> each simulation has its own job and runs parallel and independent to the others. Adding a new case, can be done in one line.

- All simulations must be set up in the same way. They need
 - ▣ an *Allrun.sh*-script, that executes all necessary steps
 - and
 - ▣ a postprocessing-JupyterNotebook, that writes its success into a *.txt*-file.

The parameter studies are set up in the same way.



- The **code presented** is from the [twophaseintertrackfoam project](https://gitlab.com/interface-tracking), which will be public or have a public mirror soon, please check for updates:
 - <https://gitlab.com/interface-tracking>
- The **official documentations** of GitHub and GitLab:
 - <https://docs.github.com/en/actions>
 - <https://docs.gitlab.com/ee/ci/>



- This presentation is available at <https://doi.org/10.5281/zenodo.10704898>.
- The **Knowledge Base** is available at <https://knowledge-base.nfdi4ing.de>.
- **Preprint** describing the workflow:
A Research Software Engineering Workflow for Computational Science and Engineering;
Marić, Gläser, Lehr et al., 2022, <https://doi.org/10.48550/arXiv.2208.07460>
- **Slides** about the workflow **including hands-on**:
FAIR research software development: A Research Software Engineering workflow;
Schwarzmeier, Marić et al., 2023, <https://doi.org/10.5281/zenodo.8333451>
- **Slides** about the workflow **at full length including hands-on**:
"Continuous" Integration of Scientific Software (in Computational Science and Engineering);
Marić et al., 2021, <https://zenodo.org/record/5522820.YnTOvnVByXI>



Interaction between
Transport and Wetting Processes

Funded by the German Research Foundation (DFG) – Project-ID 265191195 –
CRC 1194 : Z-INF



NFDI4ing

Funded by the German Research Foundation (DFG) – Project-ID 442146713 –
NFDI4ing : Base Service S-2 *Research software development*