

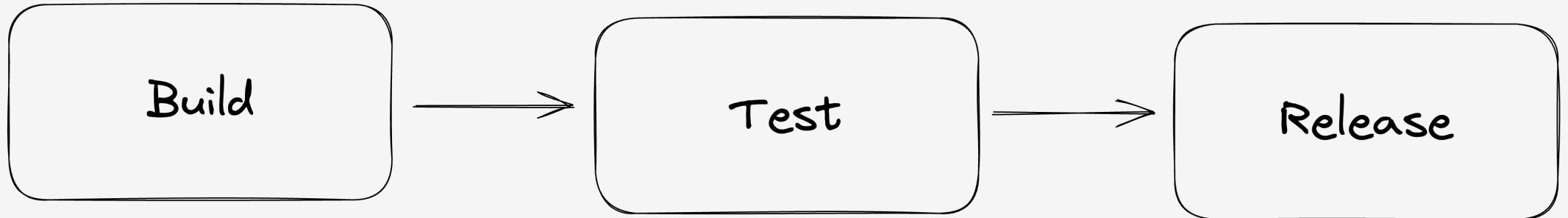
PyCon Poland 2023

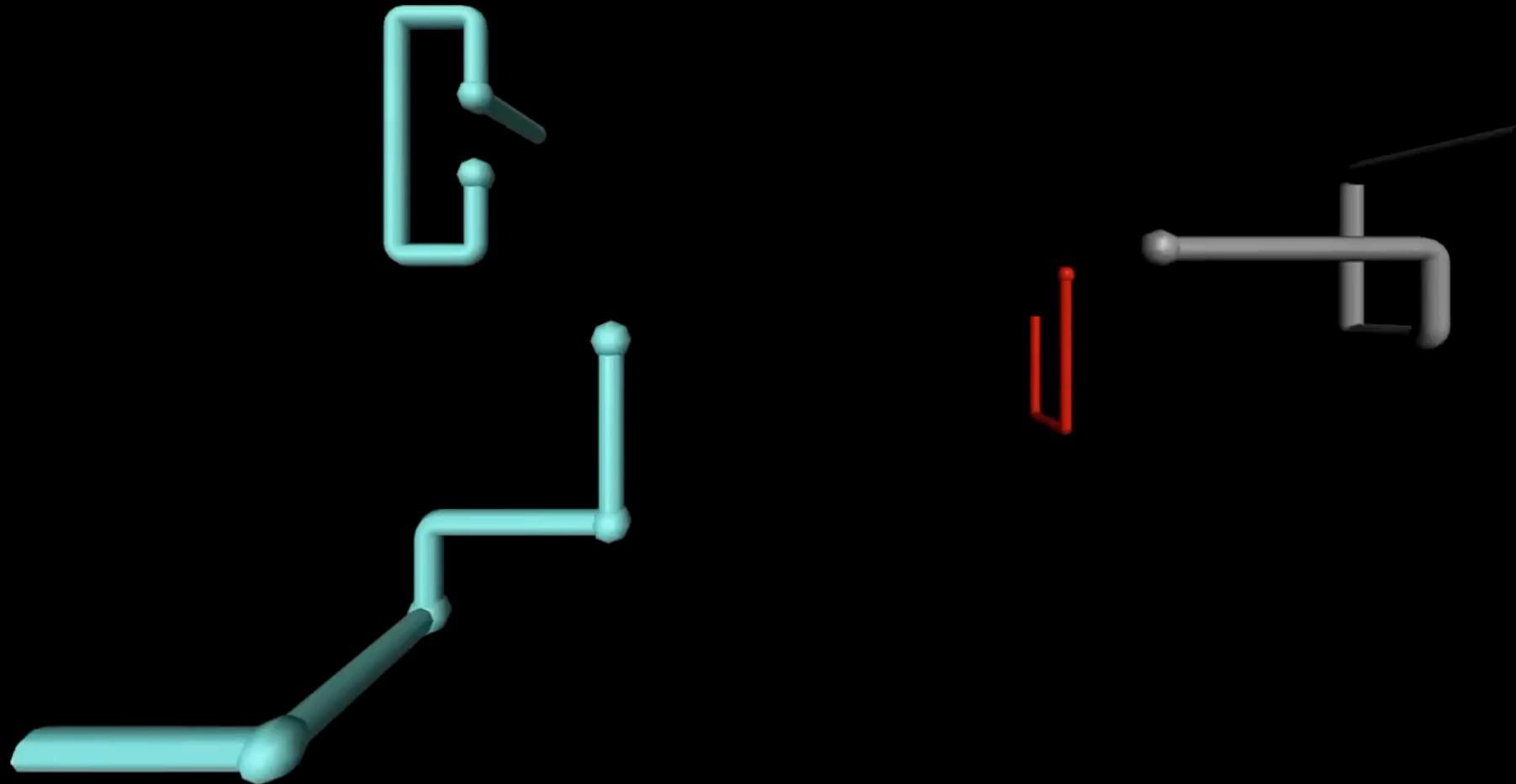
# Optimizing Your CI Pipelines

---

Sebastian Witowski

# Simple pipelines are *relatively* easy





# Roadmap

Better Docker setup



Make things run faster



Run less often, stop fast



Tips & tricks

# Choosing a CI system

# Choosing a CI system



Posts

← r/ r/devops

Posted by u/Leveronni 8 months ago

52 What is your preferred CI/CD tool?

What is your preferred CI/CD tool? If not in the list, say what you use in the comments.

Edit: My bad for not putting Azure Devops on here, I didnt see it in other threads or on the web. But that is what the "other" selection is for and comments section. Thank you!

3.7k votes

1.1k	Gitlab CI/CD
578	Jenkins
721	Github Actions
107	Circle CI
72	TeamCity
1.2k	Other / I want to see the answers

Voting closed 8 months ago

111 Comments Share Save Hide Report 90% Upvoted

# Example project

<https://gitlab.com/switowski/optimizing-ci-pipelines/-/tree/start>

The screenshot shows a GitLab project page for 'Optimizing CI pipelines'. The project was created by Sebastian Witowski (Project ID: 45658342). It has 2 commits, 5 branches, 0 tags, 1.6 MB of project storage, and 1 environment. A recent commit by Sebastian Witowski (32307cec) updated the README. The repository path is master / optimizing-ci-pipelines /. The README.md file contains the following content:

```
A dummy Django project for my CI talk

This is a dummy Django project created from https://github.com/jefftriplett/django-startproject with a simple TODO application.

Its only purpose is to be used as an example app for my "Optimizing Your CI Pipelines" talk.

If you're reading this after the summer of 2023, you can probably find my talk on Youtube. If you're reading it earlier, come to PyCon Italia or EuroPython to see it live 😊.

This is NOT production-ready (e.g., DB is configured with "POSTGRES_HOST_AUTH_METHOD=trust" setting to allow anyone to access).

Structure of this project

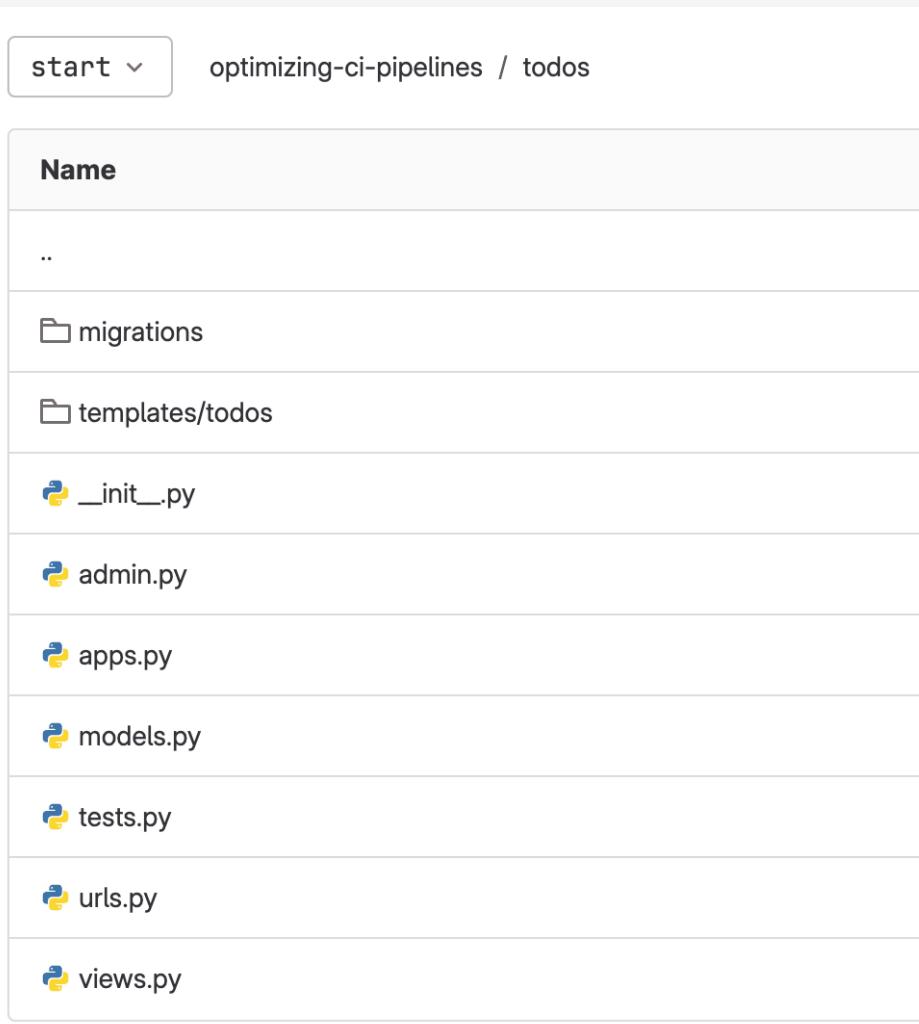
The most important parts of this project are the following:



- todo folder contains a simple TODO app. It's ugly, but it interacts with the database, and we have some tests that check if adding or completing tasks works correctly. So that's enough to simulate a small application.
- requirements.in (and in turn requirements.txt - run make deps to update requirements.txt based on the content of requirements.in file) contains

```

# Example project



# Example project

The screenshot shows a file browser interface with a sidebar on the left and a main content area on the right.

**Sidebar (Left):**

- start ▾
- optimizing-ci-
- Name
- ..
- 📁 migrations
- 📁 templates/todos
- 🐍 \_\_init\_\_.py
- 🐍 admin.py
- 🐍 apps.py
- 🐍 models.py
- 🐍 tests.py
- 🐍 urls.py
- 🐍 views.py

**Main Content Area (Right):**

File: requirements.in 576 bytes

```
1 # Django requirements
2 django-click
3 Django<4.2
4 environs[django]
5 psycopg2-binary
6 whitenoise
7 django-extensions
8 ipython
9
10 django-test-plus
11 model-bakery
12 pre-commit
13 pytest
14 pytest-cov
15 pytest-django
16 pytest-test-groups
17
18 # Bunch of dummy Python packages only meant to slow down the pip install process
19 astropy
20 bokeh
21 breathe
22 certifi
23 coverage
24 cython
25 dask
26 matplotlib
27 numpy
28 numpydoc
29 packaging
```

# Example project

start ▾ optimizing-ci-

Name

..

📁 migrations

📁 templates/todos

🐍 \_\_init\_\_.py

🐍 admin.py

🐍 apps.py

🐍 models.py

🐍 tests.py

🐍 urls.py

🐍 views.py

🐍 test\_sleeping.py 370 bytes

```
1 # Dummy tests that are slow because they sleep a lot
2
3 import time
4 import pytest
5
6
7 def test_slow_addition():
8     time.sleep(5)
9     assert 2 + 2 == 4
10
11 def test_slow_subtraction():
12     time.sleep(3)
13     assert 10 - 5 == 5
14
15
16 @pytest.mark.parametrize("sleep_time", [1, 2, 3, 4, 5])
17 def test_slow_multiplication(sleep_time):
18     time.sleep(sleep_time)
19     assert 3 * 5 == 15
20
21
22
23
24
25
26
27
28
29
```

# Example project

## Dockerfile 380 bytes

```
1 FROM python:3.10-slim-buster
2
3 # Don't write .pyc files
4 ENV PYTHONDONTWRITEBYTECODE 1
5 # Don't buffer stdout and stderr output
6 ENV PYTHONUNBUFFERED 1
7
8 COPY requirements/requirements.txt /tmp/requirements.txt
9
10 RUN pip install --upgrade pip pip-tools && \
11     pip install -r /tmp/requirements.txt
12
13 COPY . /src/
14
15 WORKDIR /src/
16
17 CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
18
```

## docker-compose.yml 780 bytes

```
1 services:
2
3 db:
4   image: "postgres:14"
5   environment:
6     - "POSTGRES_HOST_AUTH_METHOD=trust"
7   volumes:
8     - postgres-data:/var/lib/postgresql/data/
9   healthcheck:
10    test: ["CMD-SHELL", "pg_isready -U postgres"]
11    interval: 5s
12    timeout: 5s
13    retries: 5
14
15
16 web:
17   build:
18     context: .
19     dockerfile: ./Dockerfile
20   command: python manage.py runserver 0.0.0.0:8000
21   depends_on:
22     db:
23       condition: service_healthy
24   environment:
25     - "ALLOWED_HOSTS=*"
26     - "DATABASE_URL=postgres://postgres@db/postgres"
27     - "DJANGO_DEBUG=true"
28   ports:
29     - "8000:8000"
30   volumes:
31     - .:/src
32     - django-media:/src/media
33     - django-static:/src/static
34
```

# Example project

Sebastian Witowski > Optimizing CI pipelines > Pipelines > #858020958

passed Pipeline #858020958 triggered 3 days ago by  Sebastian

## Initial version

⌚ 6 jobs for **start**

in 5 minutes and 49 seconds, using 5.96 compute credits, and was q

⤟ latest

⤠ [be4f4d58](#) ⌚

⤠ No related merge requests found.

Pipeline    Needs    Jobs 6    Tests 0

build

test

deploy

 build 

 test 

 deploy 

1:34m

3:21m

0:54m

```
...  
build:  
  stage: build  
  script:  
    - docker compose build  
  
test:  
  stage: test  
  script:  
    - docker compose run --rm web python manage.py migrate  
    - docker compose run --rm web pytest  
  
deploy:  
  stage: deploy  
  script: echo "Here goes deployment script"  
  environment: production
```

*Certificate of*



**ABSOLUTELY NOT  
PRODUCTION GRADE!**

# Docker

---



# Optimize your Docker configs!

Layers caching, tags, etc.

# Improve your Docker config

```
FROM python:3.10-slim-buster
```

```
FROM python:3.10-alpine
```

# Improve your Docker config

```
FROM python:3.10-slim-buster
```

**Bigger image**

**Shorter build time**

```
FROM python:3.10-alpine
```

**Smaller image**

**Longer build time**

# Push (pull) image to (from) registry

```
...  
build:  
  stage: build  
  script:  
    - docker compose build  
  
test:  
  stage: test  
  script:  
    - docker compose run --rm web python manage.py migrate  
    - docker compose run --rm web pytest
```

# Push (pull) image to (from) registry

```
...
before_script:
- docker login -u $CI_REGISTRY_USER -p $CI_REGISTRY_PASSWORD $CI_REGISTRY

build:
  stage: build
  script:
    - docker compose build
    - docker push registry.gitlab.com/switowski/optimizing-ci-pipelines/web:dev

test:
  stage: test
  script:
    - docker pull registry.gitlab.com/switowski/optimizing-ci-pipelines/web:dev
    - docker compose run --rm web python manage.py migrate
    - docker compose run --rm web pytest
```

# Push (pull) image to (from) registry

```
...  
before_script:  
- docker login -u $CI_REGISTRY_USER -p $CI_REGISTRY_PASSWORD  
  
build:  
  stage: build  
  script:  
    - docker compose build  
    - docker push registry.gitlab.com/  
  
test:  
  stage: test  
  script:  
    - docker pull registry.gitlab.com/  
    - docker compose run --rm web  
    - docker compose run --rm web
```

## Push and pull docker image to container registry

⌚ 3 jobs for **push-pull-docker-image** ← **Branch name**  
in 6 minutes and 2 seconds, using 6.04 compute credits, and was queued for 1 second

latest

9ab8ea6c ↗

No related merge requests found.

Pipeline   Needs   Jobs 3   Tests 0

build	test	deploy	Branch
<span>✓</span> build <span>↻</span> 1:34m	<span>✓</span> test <span>↻</span> 3:21m	<span>✓</span> deploy <span>↻</span> 0:54m	<span>start</span>
<span>1:46m</span>	<span>3:14m</span>	<span>1:01m</span>	<span>push-pull-docker-image</span>

# Multistage builds

Builder image that ends up large

```
FROM small-image AS builder  
COPY (bunch of files needed ONLY for the build process)  
.....  
apt-get install (a lot of dependencies)  
.....  
RUN (build process that creates a lot of leftover files)  
.....
```

Production image that ends up small

```
FROM small-image AS prod  
COPY --from=builder a-small-binary .  
RUN a-small-binary
```



Make things run  
**faster**

# Make things run faster (in parallel)

GitLab.org • GitLab • Pipelines • #856315395

(running) Pipeline #856315395 triggered 30 minutes ago by Kristina Doskic

Ruby 3.0 merged\_result MR pipeline

83 jobs for f119085 with move-github-import-pull-request-reviews into master

Merge branch 'move-github-import-pull-request-reviews' into 'master' b81baef60

1 related merge request: f119085 GitHub Import: Move PR reviews under pull\_requests namespace

Pipeline Needs Jobs 83 Tests 81 Code Quality

Group jobs by Stage Job dependencies

sync

- don't-interrupt-me
- no-jh-check
- qa:selectors
- rails-production-server-boot
- compile-production-assets
- compile-production-assets-as-if-foss
- compile-test-assets
- compile-test-assets-as-if-foss
- detect-previous-failed-tests
- detect-tests
- e2e-test-pipeline-generate
- retrieve-frontend-fixtures
- retrieve-tests-metadata
- review-cleanup
- review-stop
- rspec-foss-impact:pipeline-generate
- rspec-predictive:pipeline-generate
- setup-test-env

preflight

- no-jh-check
- qa:selectors
- rails-production-server-boot

prepare

- compile-production-assets
- compile-production-assets-as-if-foss
- compile-test-assets
- compile-test-assets-as-if-foss
- detect-previous-failed-tests
- detect-tests
- e2e-test-pipeline-generate
- retrieve-frontend-fixtures
- retrieve-tests-metadata
- review-cleanup
- review-stop
- rspec-foss-impact:pipeline-generate
- rspec-predictive:pipeline-generate
- setup-test-env

build-images

- build-assets-image
- build-assets-image-as-if-foss
- build-qa-image
- build-qa-image-as-if-foss

fixtures

- graphql-schema-dump
- rspec-all-frontend\_fixture

lint

- brakeman-sast
- code\_quality
- docs code\_quality
- eslint
- feature-flags-usage
- generate-apollo-graphql-schema
- haml-lint
- haml-lint ee
- lint-yaml
- ping-appsec-for-sast-findings
- rubocop
- secret\_detection
- semgrep-appsec-custom-rules
- semgrep-sast
- static-analysis
- static-verification-with-database

test

- compile-storybook
- danger-review
- db:check-schema
- gitlab\_glt\_test
- graphql-verify
- jest predictive
- jest-integration
- memory-on-boot
- rspec fail-fast
- rspec fast\_spec\_helper
- rspec rspec-ee-pg13-rerun-previous-failed-te...
- rspec rspec-ee-pg13-rerun-previous-failed-tests
- rspec-ee fail-fast
- rspec-ee:predictive:trigger
- rspec-foss-impact:trigger
- rspec:predictive:trigger
- run-dev-fixtures
- run-dev-fixtures-ee
- startup-css-check
- verify-tests-yml
- webpack-dev-server

post-test

- coverage-frontend
- fail-pipeline-early
- rspec.coverage
- rspec:feature-flags
- rspec:flaky-tests-report

review

- start-review-app-pipeline

qa

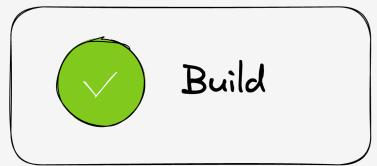
- e2e:package-and-test-ce
- e2e:package-and-test-ee
- e2e:test-on-gdk

Downstream

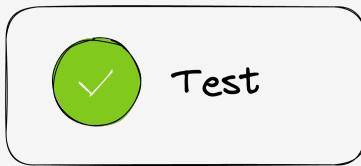
- rspec-ee:predictive... #856329484 Child
- rspec:predictive... #856328576 Child
- e2e:test-on-gdk #856315501 Child

# Make things run faster (in parallel)

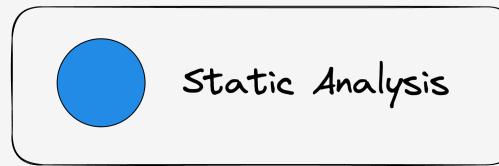
Build stage



Test stage



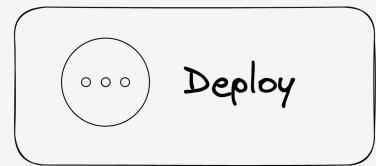
Static analysis stage



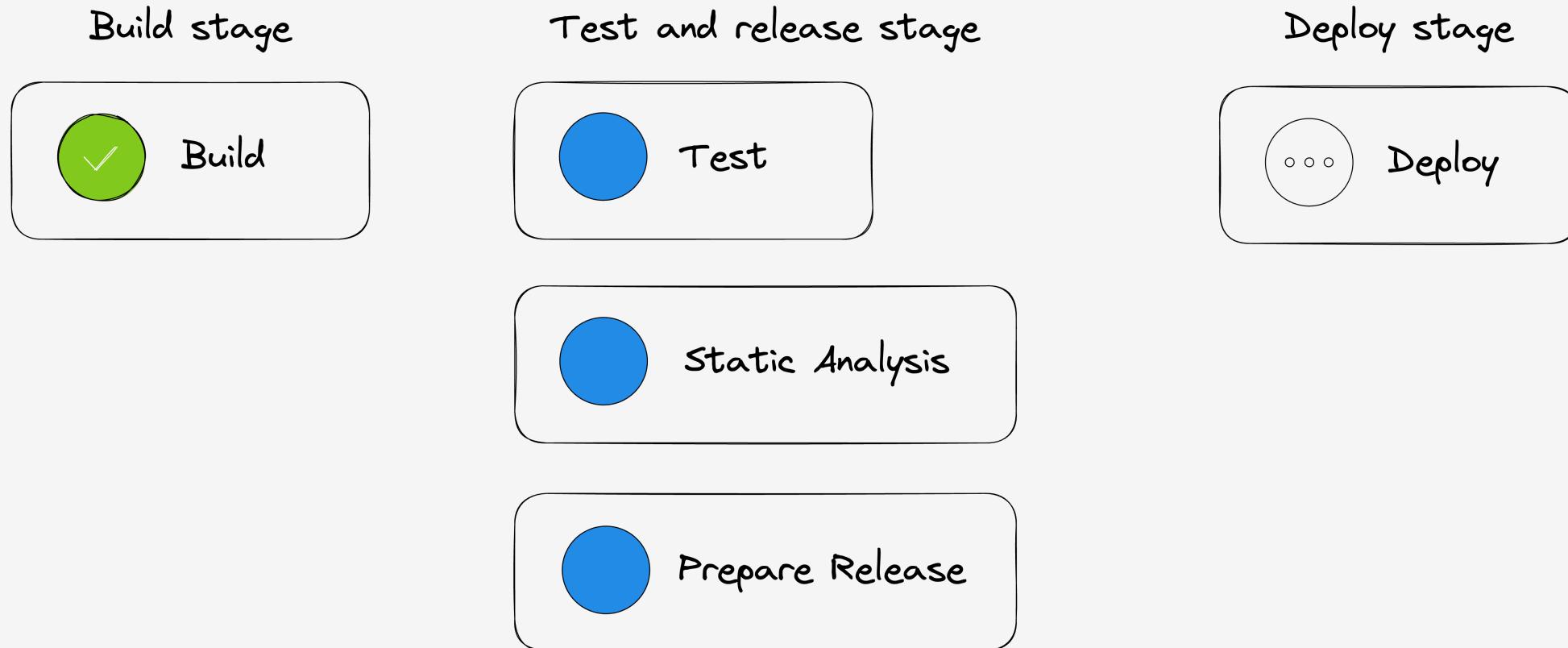
Release stage



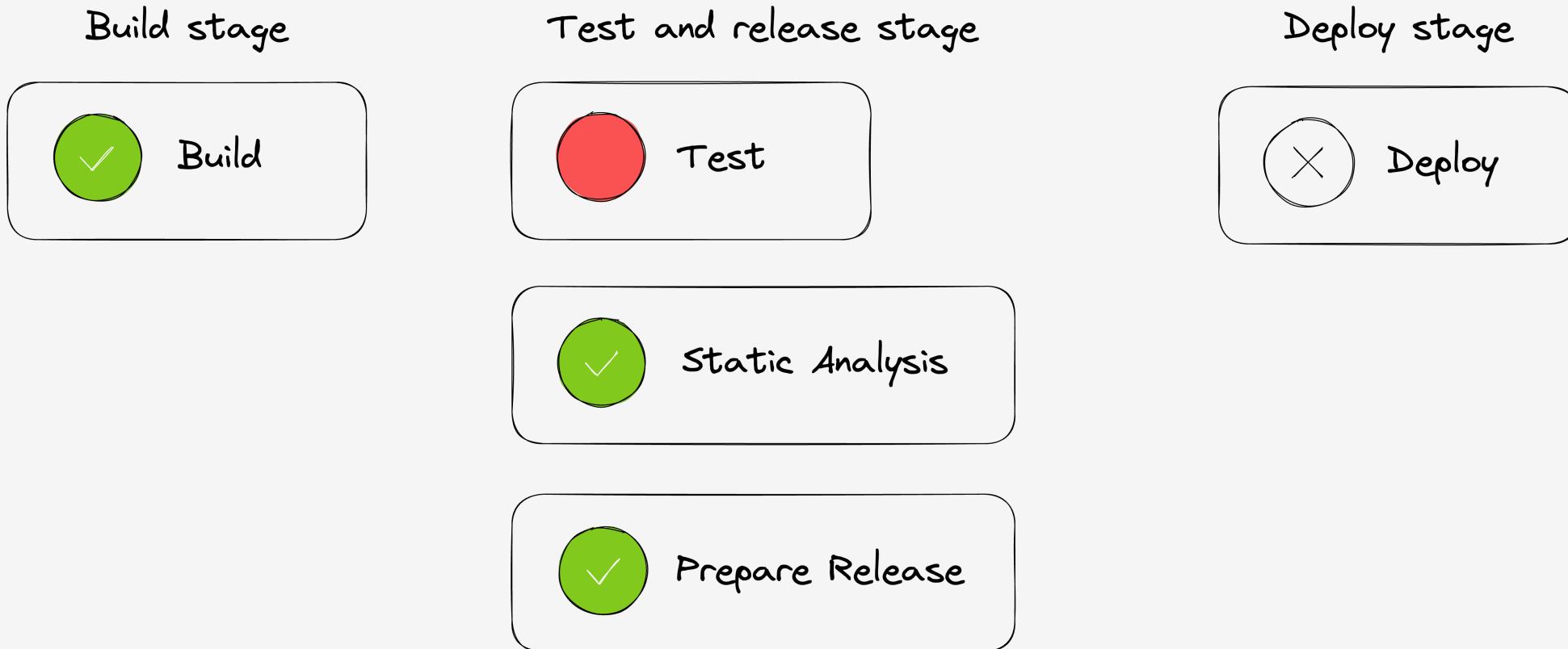
Deploy stage



# Make things run faster (in parallel)



# Make things run faster (in parallel)



There is an [open issue](#) about this from 2018.

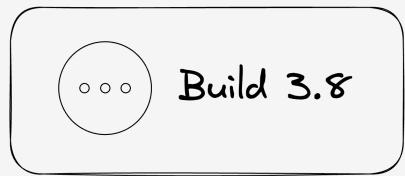
# Directed Acyclic Graph

One job **starts** after  
another **finishes**,  
**regardless** of what  
stage they are in.

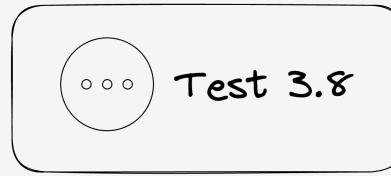


# Without DAG

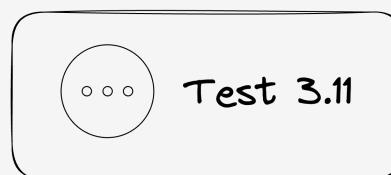
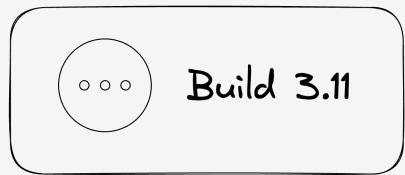
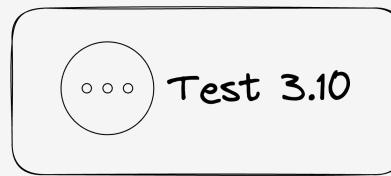
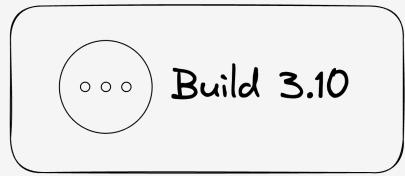
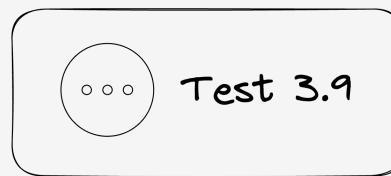
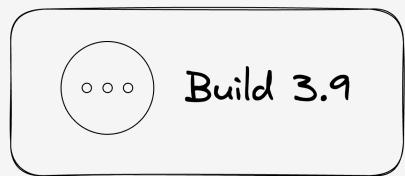
Build stage



Test stage

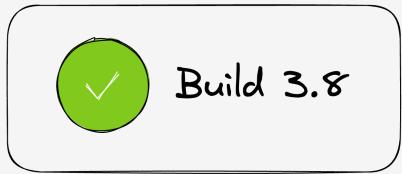


Release stage

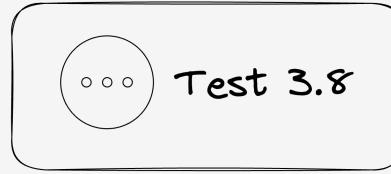


# Without DAG

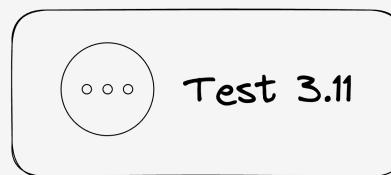
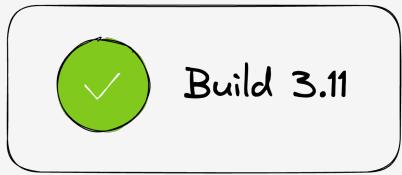
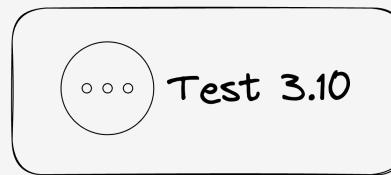
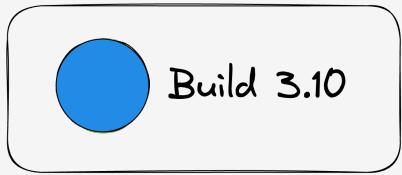
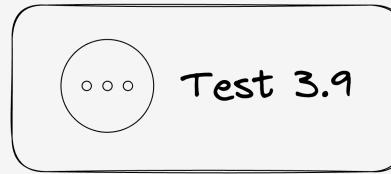
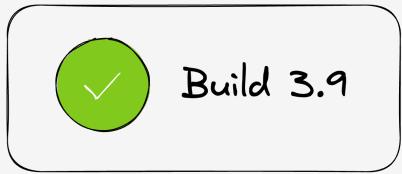
Build stage



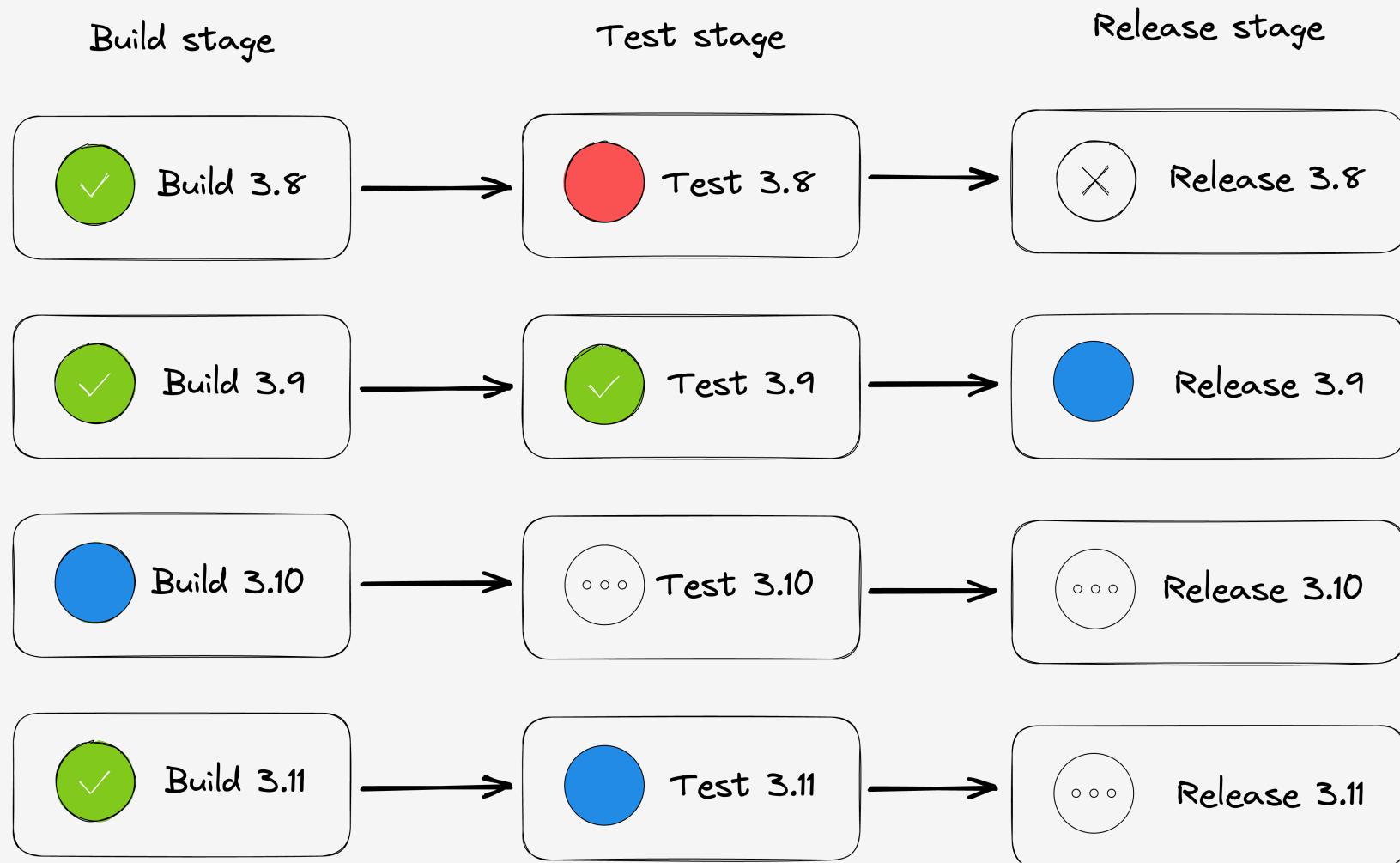
Test stage



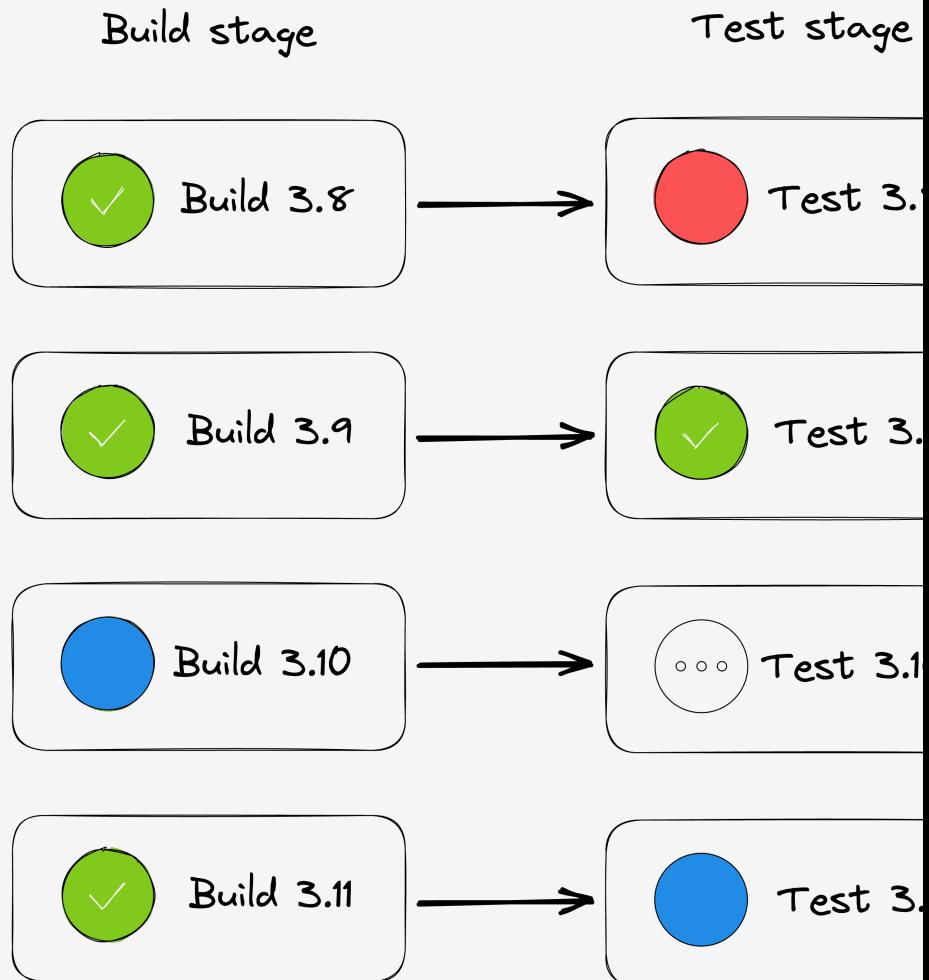
Release stage



# With DAG



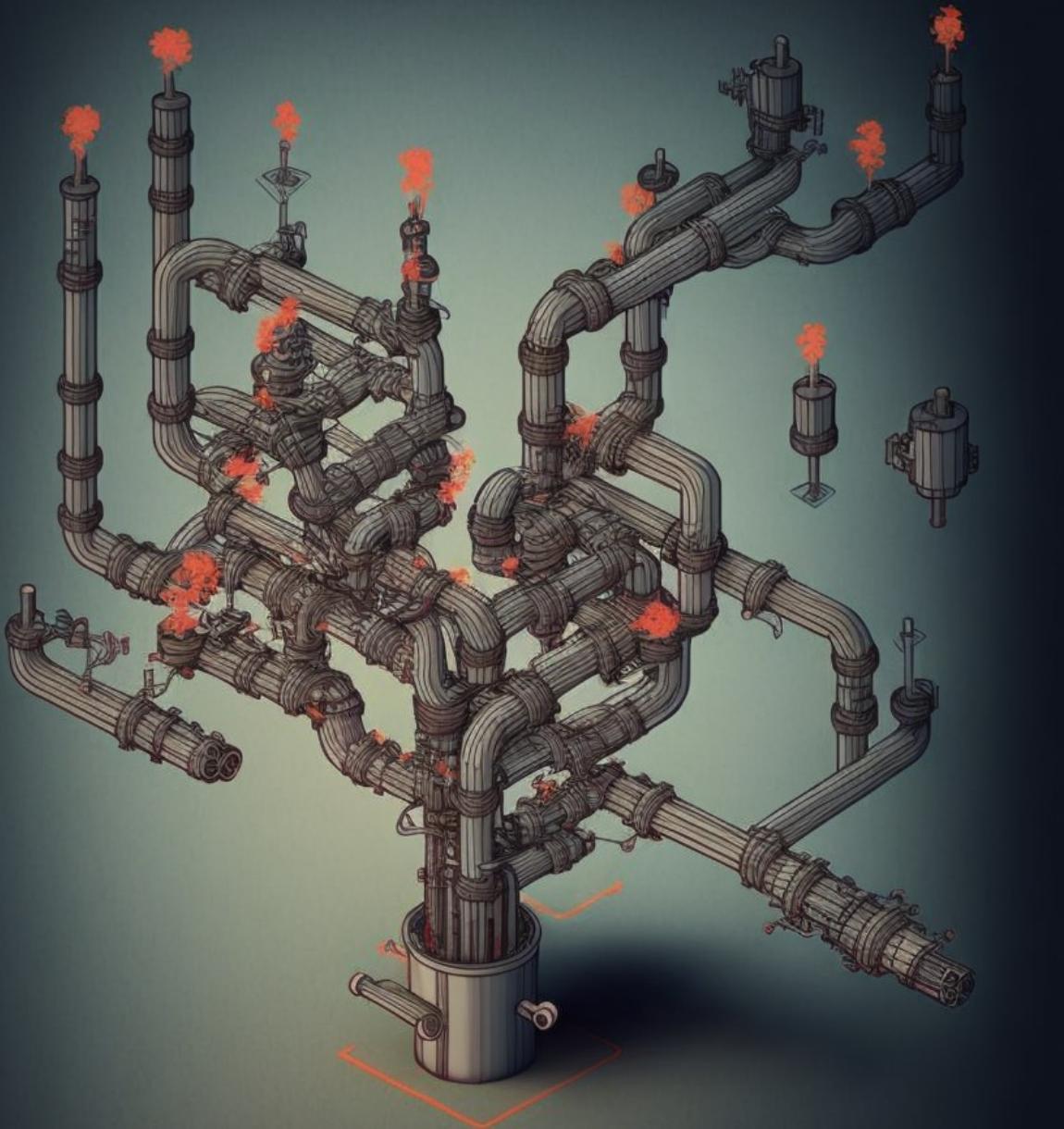
# With DAG



```
build_3.8:
stage: build
script:
- docker build -f Dockerfile_3.8
- docker push $CI_REGISTRY_IMAGE:3.8

test_3.8:
stage: test
needs: ["build_3.8"]
script:
- docker pull $CI_REGISTRY_IMAGE:3.8
- docker run $CI_REGISTRY_IMAGE:3.8 pytest

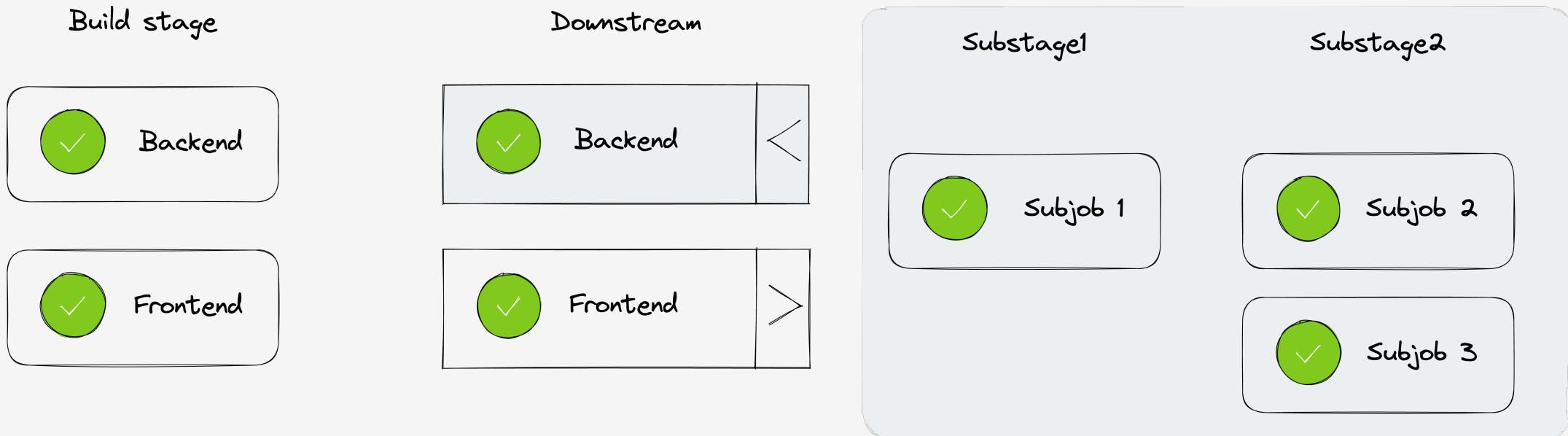
release_3.8:
stage: release
needs: ["test_3.8"]
script: echo "Release script"
```



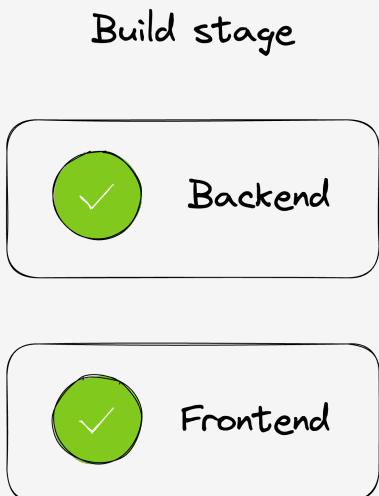
# Downstream (child) pipeline

Separate mini-pipelines  
that can be triggered  
from your main pipeline.

# Downstream (child) pipelines



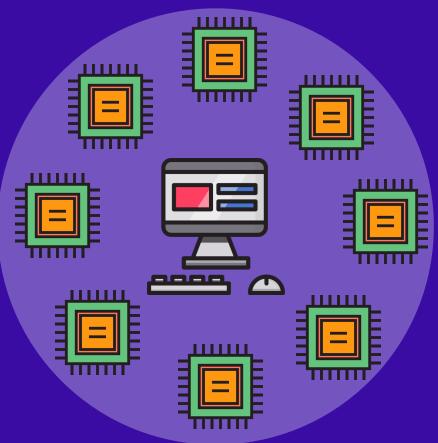
# Downstream (child) pipelines



```
frontend:  
  trigger:  
    include: frontend/.gitlab-ci.yml  
    strategy: depend  
  rules:  
    - changes: [frontend/*]  
  
backend:  
  trigger:  
    include: backend/.gitlab-ci.yml  
    strategy: depend  
  rules:  
    - changes: [backend/*]
```

# Run tests in parallel

# Run tests in parallel

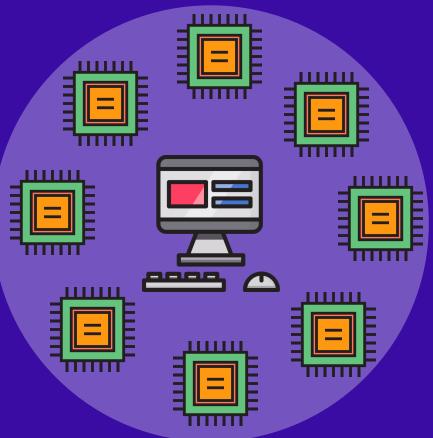


**pytest-xdist**

Run tests across  
multiple **CPUs**

```
$ pip install pytest-xdist  
$ pytest -n auto
```

# Run tests in parallel



## **pytest-xdist**

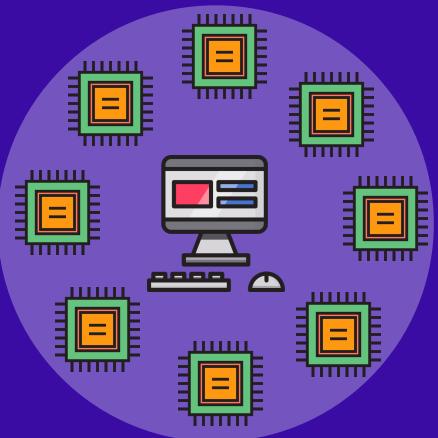
Run tests across  
multiple **CPUs**



## **pytest-test-groups**

Run tests across  
multiple **runners**

# Run tests in parallel



## pytest-xdist

Run tests across multiple **CPUs**

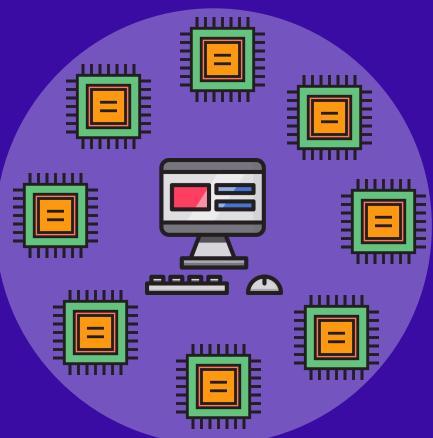


## pytest-test-groups

Run tests across multiple **runners**

```
# requirements.in  
pytest-test-groups  
  
# .gitlab-ci.yml  
test:  
  stage: test  
  parallel: 5  
  script:  
    - pytest \  
      --test-group-count $CI_NODE_TOTAL \  
      --test-group=$CI_NODE_INDEX
```

# Run tests in parallel



## pytest-xdist

Run tests across multiple **CPUs**



## pytest-test-groups

Run tests across multiple **runners**

## Running pytest in groups

⌚ 7 jobs for **parallel-pytest-in-groups** ← **Branch name**  
in 4 minutes and 56 seconds, using 12.24 compute credits, and was queued for 0 seconds

latest

8d01c47e ↗

No related merge requests found.

Pipeline   Needs   Jobs 7   Tests 0

build	test	deploy
build	5	test 1/5 test 2/5 test 3/5 test 4/5 test 5/5

# Run **less** and stop **fast**

- Interruptible jobs

General pipelines    **Settings -> CI/CD -> General pipelines**

Customize your pipeline configuration.

- Public pipelines  
Allow public access to pipelines and job details, including output logs and artifacts. [?](#)
- Auto-cancel redundant pipelines  
Pipelines for new changes cause older pending or running pipelines on the same branch to be cancelled. [?](#)
- Prevent outdated deployment jobs  
When a deployment job is successful, prevent older deployment jobs that are still pending. [?](#)

# Run **less** and stop **fast**

- Interruptible jobs

General pipelines    **Settings -> CI/CD -> General pipelines**

Customize your pipeline configuration.

- Public pipelines  
Allow public access to pipelines and job details, including output logs and artifacts. [?](#)
- Auto-cancel redundant pipelines  
Pipelines for new changes cause older pending or running pipelines on the same branch to be cancelled. [?](#)
- Prevent outdated deployment jobs  
When a deployment job is successful, prevent older deployment jobs that are still pending. [?](#)

```
build:  
  stage: build  
  interruptible: true  
  script:  
    - docker compose build
```

# Run **less** and stop **fast**

- Interruptible jobs

General pipelines    **Settings -> CI/CD -> General pipelines**

Customize your pipeline configuration.

- Public pipelines  
Allow public access to pipelines and job details, including output logs and artifacts. [?](#)
- Auto-cancel redundant pipelines  
Pipelines for new changes cause older pending or running pipelines on the same branch to be cancelled. [?](#)
- Prevent outdated deployment jobs  
When a deployment job is successful, prevent older deployment jobs that are still pending. [?](#)

```
build:  
  stage: build  
  interruptible: true  
  script:  
    - docker compose build
```

- Stop fast (pytest -x)

**Tip 5:**  
Not running  
things in the CI

Not every check is  
**mandatory** in the CI

# Not every check is **mandatory** in the CI

Some can run only on the main branches.

Others can be triggered manually.



# Random Tips & Tricks

---

# Caching and cache policies

You can use caching to, well, cache stuff between jobs.

But you can also specify if you want to push or pull stuff to cache using **policy** key.

```
default:  
  cache: &global_cache  
  key: $CI_COMMIT_REF_SLUG  
  paths:  
    - .cache/pip  
    - some/other/path/  
  policy: pull-push  
  
job:  
  cache:  
    # inherit all global cache settings  
    <<: *global_cache  
    # override the policy  
  policy: pull
```

# Fast zip

For caching/artifacts, you can choose **different level of compression** (low level of compression runs faster, but results in a larger zip file).

```
variables:  
  FF_USE_FASTZIP: "true"  
  # Available options are:  
  # fastest, fast, default, slow,  
  # or slowest  
  ARTIFACT_COMPRESSION_LEVEL: "fastest"  
  CACHE_COMPRESSION_LEVEL: "fastest"
```

# Different builders



buildah



Kaniko

# Use your own runners

## Runner config

⌚ 3 jobs for **runner** <-- Branch name  
in 10 minutes and 28 seconds, using 0 compute credits, and was queued for 3 seconds

Newest

-o 386a7e10 📂

No related merge requests found.

Pipeline   Needs   Jobs 3   Tests 0

Status	Job	Stage	Name	Duration	Coverage
<span>✔ passed</span>	#4256431075 ↳ runner -o 386a7e10 runner	deploy	deploy	⌚ 00:00:25 22 hours ago	
<span>✔ passed</span>	#4256431074 ↳ runner -o 386a7e10 runner	test	test	⌚ 00:09:18 22 hours ago	
<span>✔ passed</span>	#4256431072 ↳ runner -o 386a7e10 runner	build	build	⌚ 00:00:44 22 hours ago	

# Use your own runners

## Runner config

⌚ 3 jobs for **runner** <-- Branch name  
in 10 minutes and 28 seconds, using 0 compute credits, and was queued for 3 seconds

Newest

-o 386a7e10 📂

No related merge requests found.

Pipeline   Needs   Jobs 3   Tests 0

Status	Job	Stage	Name	Duration	Coverage
passed	#4256431075 ↳ runner -o 386a7e10 runner	deploy	deploy	⌚ 00:00:25 22 hours ago	
passed	#4256431074 ↳ runner -o 386a7e10 runner	test	test	⌚ 00:09:18 22 hours ago	
passed	#4256431072 ↳ runner -o 386a7e10 runner	build	build	⌚ 00:00:44 22 hours ago	

# Takeaways

# Takeaways

- Learn concepts, not tools

# Takeaways

- **Learn concepts, not tools**
- **There are no silver bullets**

python-alpine or python-debian? Pull an image or build it?

# Takeaways

- **Learn concepts, not tools**
- **There are no silver bullets**

python-alpine or python-debian? Pull an image or build it?

- **Not every check has to run in every pipeline**

make MR pipelines *fast* and main branch pipelines *thorough*

# Takeaways

- **Learn concepts, not tools**
- **There are no silver bullets**  
python-alpine or python-debian? Pull an image or build it?
- **Not every check has to run in every pipeline**  
make MR pipelines *fast* and main branch pipelines *thorough*
- **Outdated CI setup is also a technical debt**

# Thank you!

[switowski.com](http://switowski.com)  
@SebaWitowski



# Attributions

Most images come from [midjourney.com](https://midjourney.com)  
Drawings were done with [excalidraw.com](https://excalidraw.com)

- Slide 1: <https://www.midjourney.com/app/search/?jobId=6f707c18-bf19-4fac-b87f-622ff3e1561b>
- Roadmap - containers: <https://www.midjourney.com/app/search/?jobId=28e92637-bfd6-4d07-ae36-a279dd35c2d4>
- Roadmap - rocket: <https://www.midjourney.com/app/search/?jobId=eaeeb022-17af-4b71-8438-a8eb05afc298>
- Roadmap - traffic lights: <https://www.midjourney.com/app/search/?jobId=89159672-debf-4845-a822-be159983f172>
- Roadmap - chest: <https://www.midjourney.com/app/search/?jobId=1bb66ddc-84a4-4ac1-a36a-7c7f8dbd1c04>
- Gitlab logo: <https://www.midjourney.com/app/search/?jobId=a5491c2d-199e-43dd-b65f-9ebefc031f65>
- Ribbon badge: [https://www.clipartmax.com/download/m2i8H7d3Z5i8G6Kg\\_certificate-ribbons-ribbon-badge-vector-png/](https://www.clipartmax.com/download/m2i8H7d3Z5i8G6Kg_certificate-ribbons-ribbon-badge-vector-png/)
- Containers: <https://www.midjourney.com/app/search/?jobId=f14ee5f1-b3a7-4bca-9907-e67607beff90>
- DAG image: <https://www.midjourney.com/app/search/?jobId=4609agaa-7154-49f2-9b10-38d600aa04f2>
- Child pipelines: <https://www.midjourney.com/app/search/?jobId=39e19a04-1196-4c2e-99e9-f242464b5d7c>
- Rube Goldberg machine: <https://www.midjourney.com/app/search/?jobId=60ab5692-a07a-4992-8641-13dfbe6193f2>

# Questions?

[switowski.com](http://switowski.com)

@SebaWitowski

<https://gitlab.com/switowski/optimizing-ci-pipelines>

Slides: <https://speakerdeck.com/switowski>