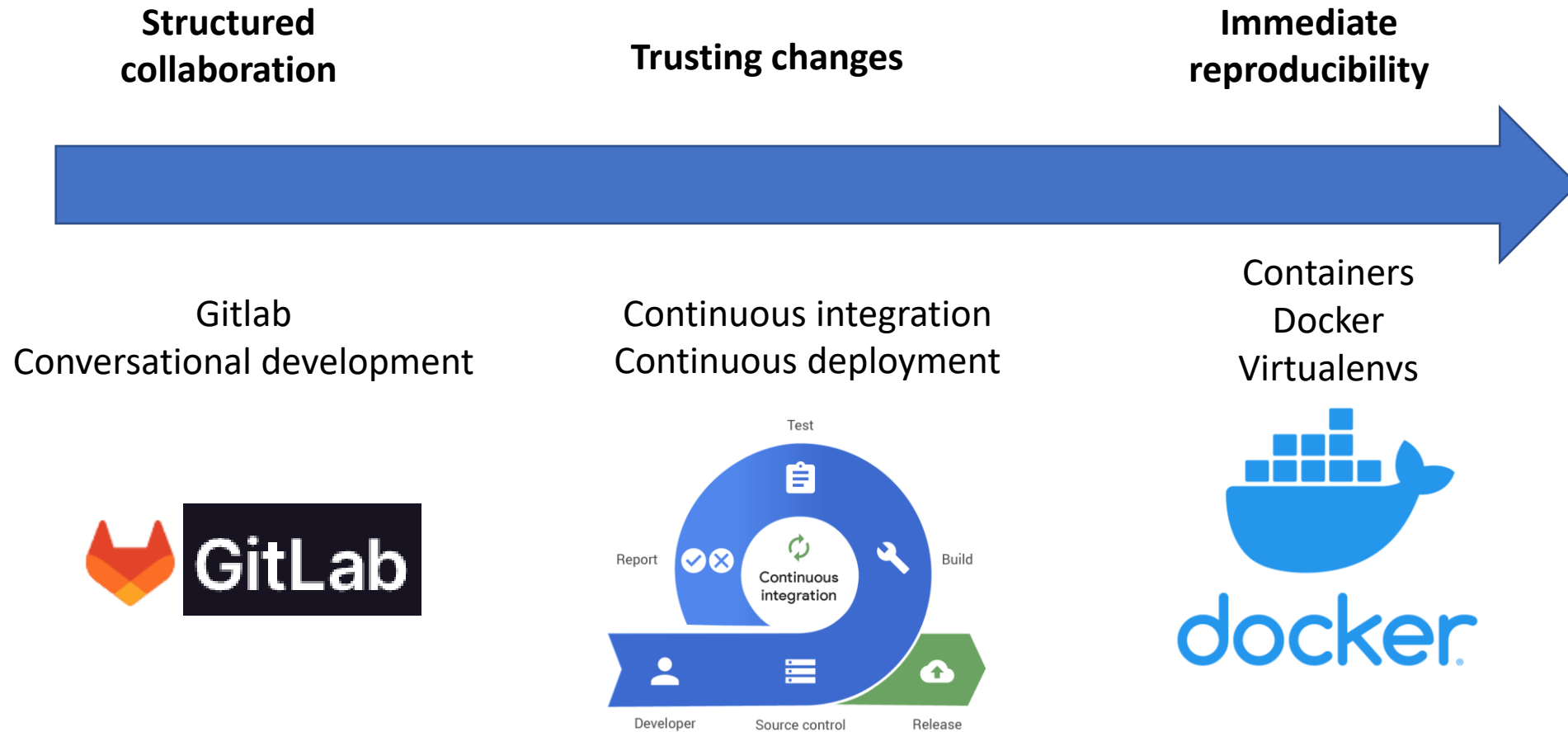


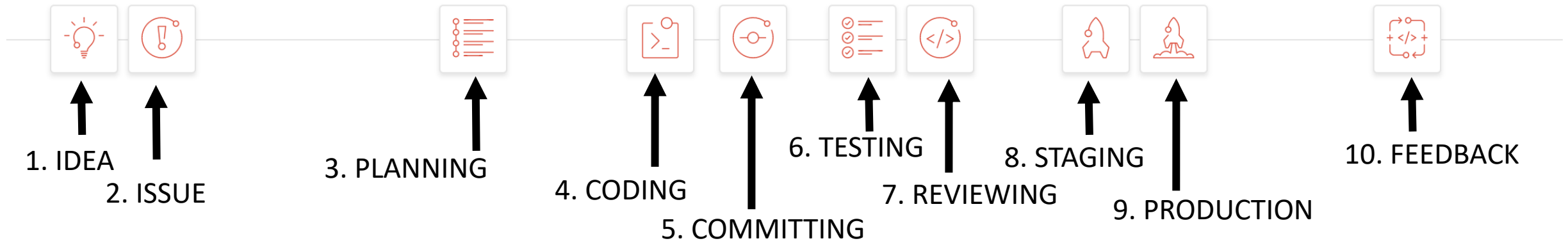
Coding: collaborative,
trustworthy, reproducible

Alessandro Corbetta

Plan of the lectures



The conversational development paradigm



Gitlab/github: meant to support this approach

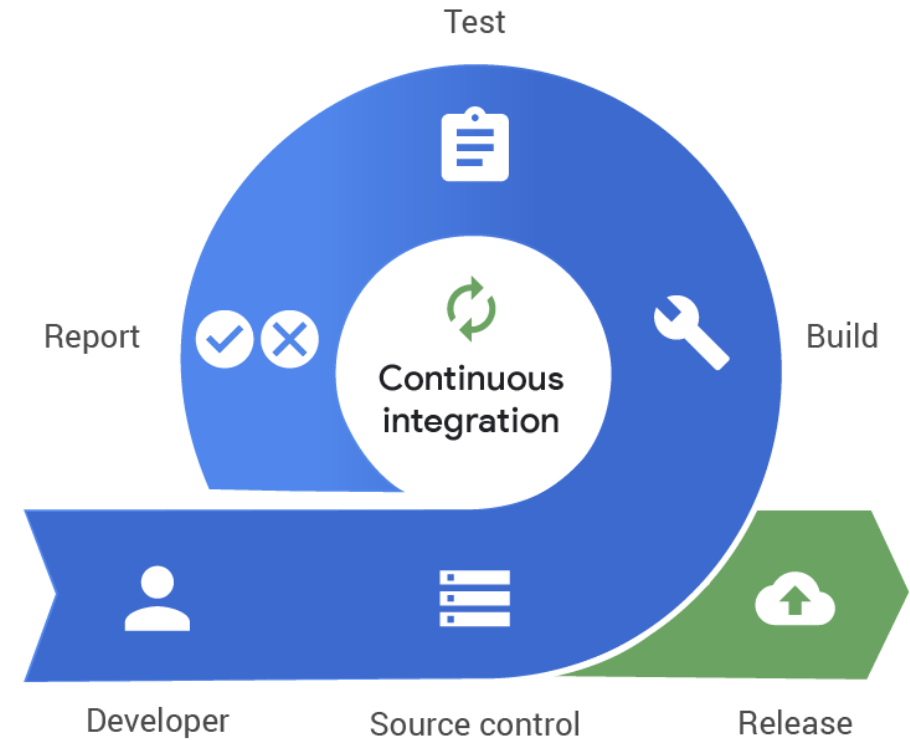
Development model => content & conversations between developers

Fosters collaborations w/o centralized entities

In my experience: very scalable also in research (codes & execution)!

Continuous integration

- **Each push: automated** remote **testing**
- Every user fully aware of the code state



- If testing quick, dev cycle & master merge: very frequent
 - Many github repos: hundreds merge per day after remote testing

Minimal python example: CI

`.gitlab-ci.yml`

```
image: python:latest
```



Running in a linux sandbox with up-to-date python (Docker container)

```
test-only:
```

```
  script:
```

Installing dependencies, here put numpy etc...
(better with requirements.txt)

```
    - pip install pytest
```



```
    - pytest -vvv
```



Testing



passed

Job test-only triggered 5 minutes ago by Administrator



test-only
test



P Pytest Example

Project information

Repository

Issues 0

Merge requests 0

CI/CD

Pipelines

Editor

Jobs

Schedules

Security & Compliance

Deployments

Packages and registries

Infrastructure

Monitor

Analytics

Wiki

Snippets

Settings

```
1 Running with gitlab-runner 15.6.1 (133d7e76)
2   on Ci runner NS9smqSy
3   ✓ Preparing the "docker" executor
4   Using Docker executor with image python:latest ...
5   Pulling docker image python:latest ...
6   Using docker image sha256:ee4e7a0f1c354d9996229a765d0785df2671252c1822ae111015d37dcf5f765b for
7   ...
8   ✓ Preparing environment
9   Running on runner-ns9smqsy-project-2-concurrent-0 via smr3696-1...
10  ...
11  ✓ Getting source from Git repository
12  Fetching changes with git depth set to 20...
13  Reinitialized existing Git repository in /builds/gitlab-instance-59615060/pytest-example/.git/
14  Checking out 2733dd46 as main...
15  ...
48  $ pytest -vvv
49  ===== test session starts =====
50  platform linux -- Python 3.11.0, pytest-7.2.0, pluggy-1.0.0 -- /usr/local/bin/python
51  cachedir: .pytest_cache
52  rootdir: /builds/gitlab-instance-59615060/pytest-example
53  plugins: cov-4.0.0
54  collecting ... collected 2 items
55  test_code.py::test_summing_f1 PASSED [ 50%]
56  test_code.py::test_summing_f2 PASSED [100%]
57  ===== 2 passed in 0.01s =====
58  ...
59  ✓ Cleaning up project directory and file based variables
60  ...
61  Job succeeded
```

Coding: collaborative,
trustworthy, **reproducible**

Alessandro Corbetta

Reproducibility

the extent to which consistent results are obtained when an experiment is repeated.



Psst..
it runs perfectly on **my** machine!



Yo, man..
Your code is broken

Yo...for real??
But which versions...

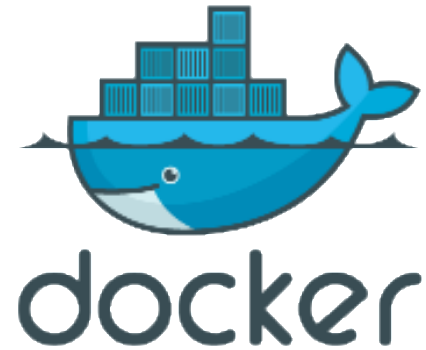
Aims

- Predictable code deployment
- Streamline execution in collaborative environment
- Reproducible science
- Byproducts:
 - Sandboxed execution
 - Preventing conflicts with system-wide dependencies

Reproducibility



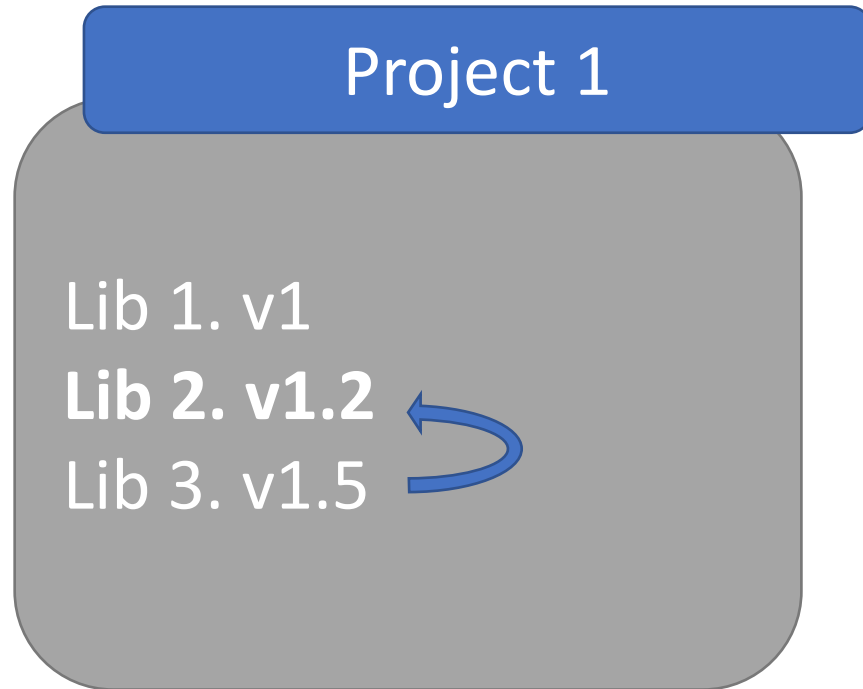
- Sandboxing python dependencies
 - Venv, virtualenv, conda env



- Light-weight sytem-level sandboxing

Python virtual environments

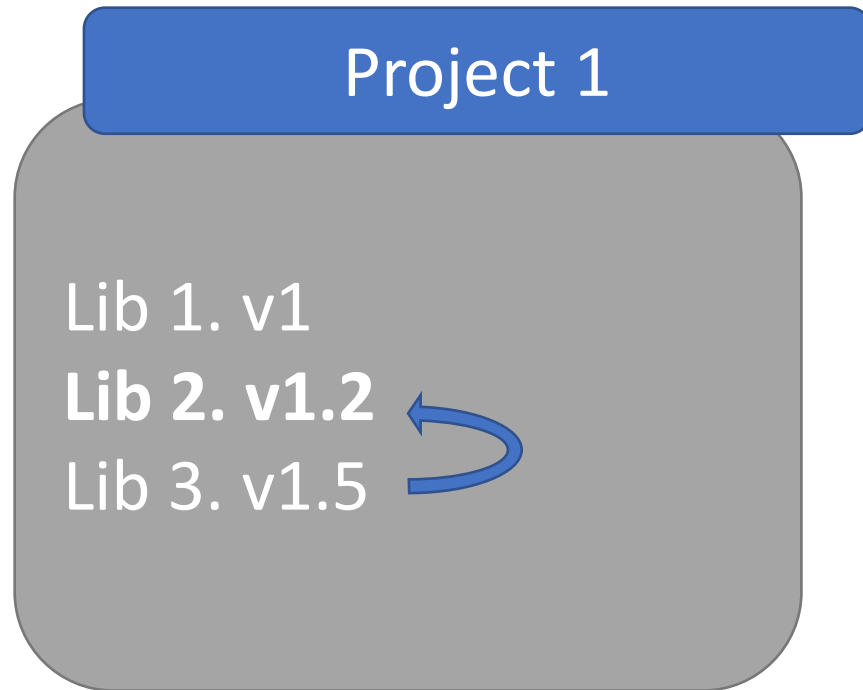
ISSUE 1: having control on the **actual project dependencies**
(and sub-dependencies)



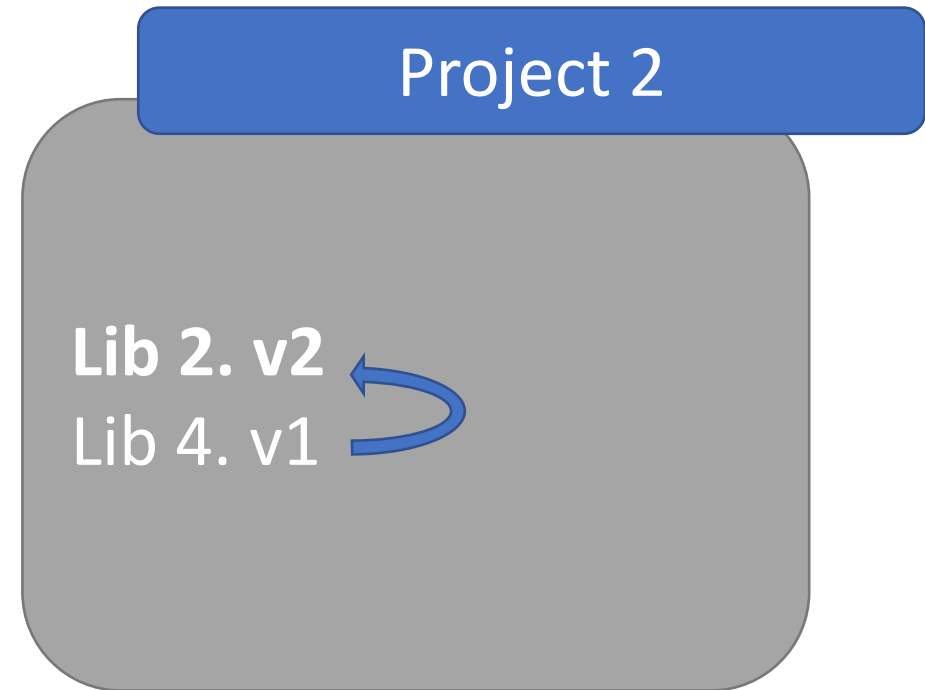
`pip install lib_2==1.2`

Python virtual environments

ISSUE 2: Allowing projects with conflicting dependencies



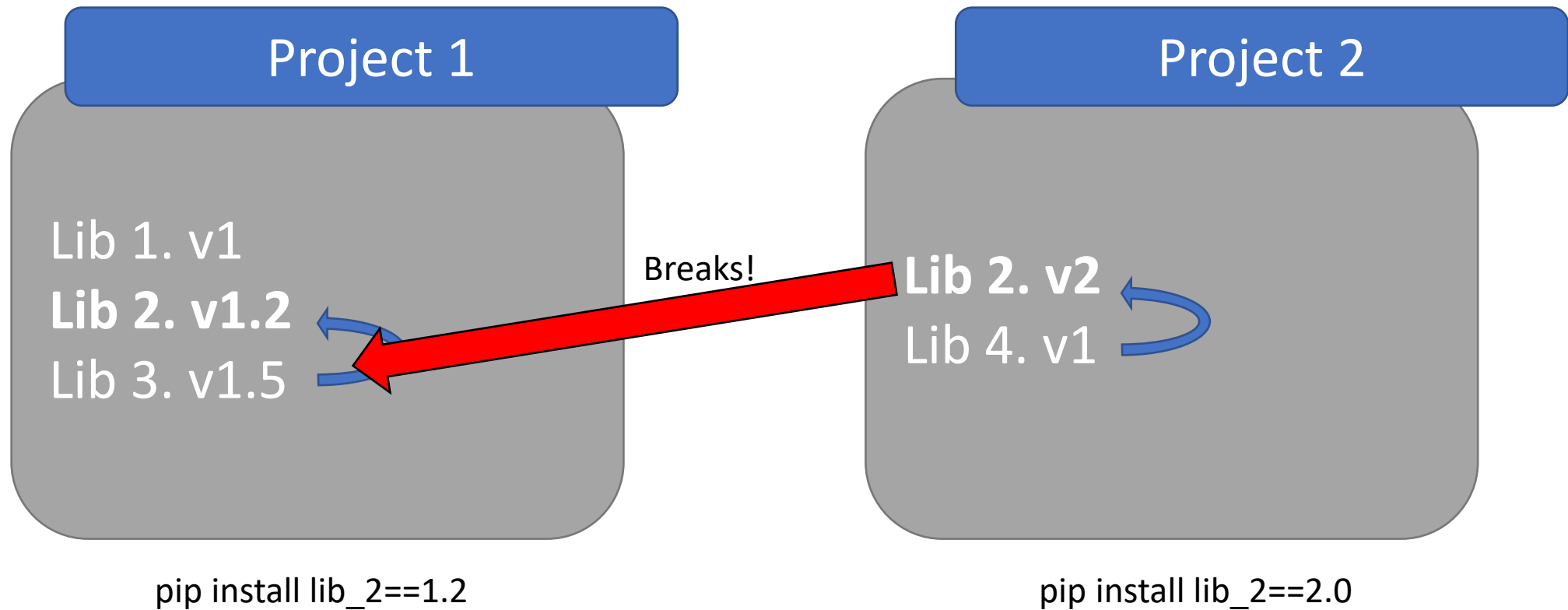
`pip install lib_2==1.2`



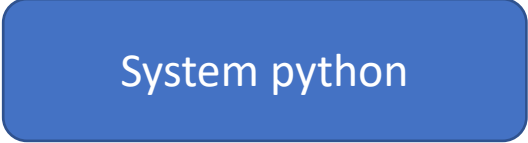
`pip install lib_2==2.0`

Python virtual environments

ISSUE 2: Allowing projects with conflicting dependencies

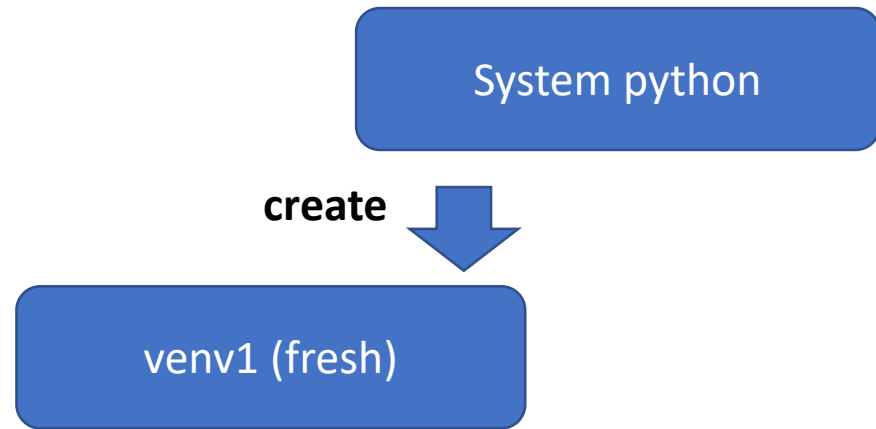


Virtual environments pattern

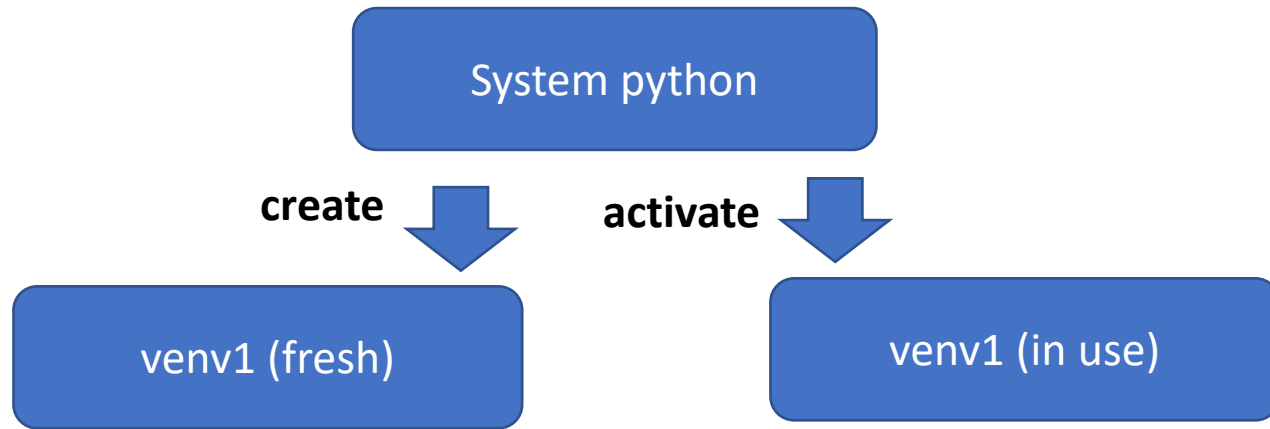


System python

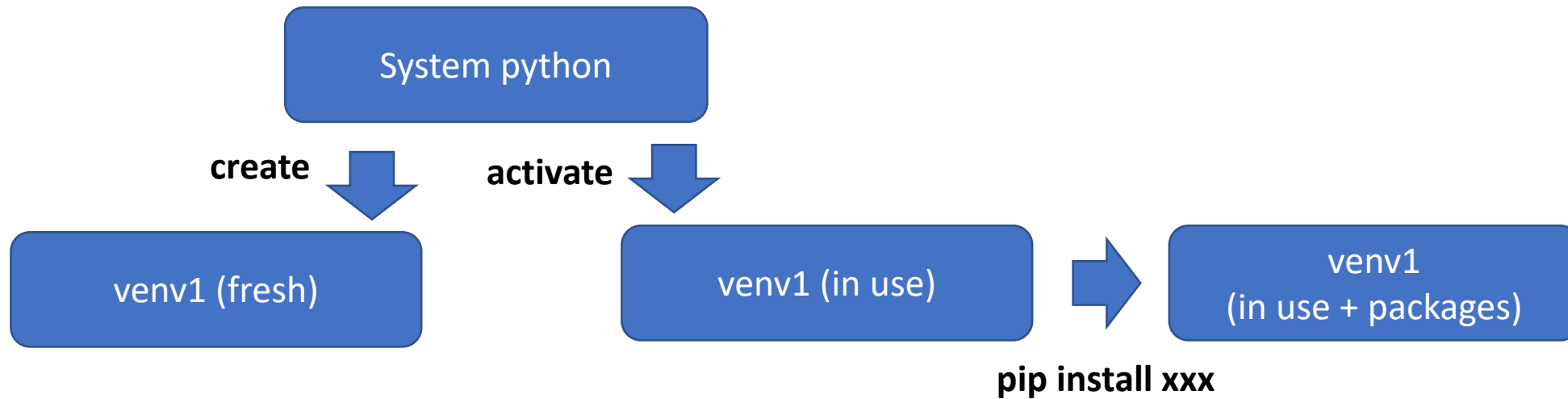
Virtual environments pattern



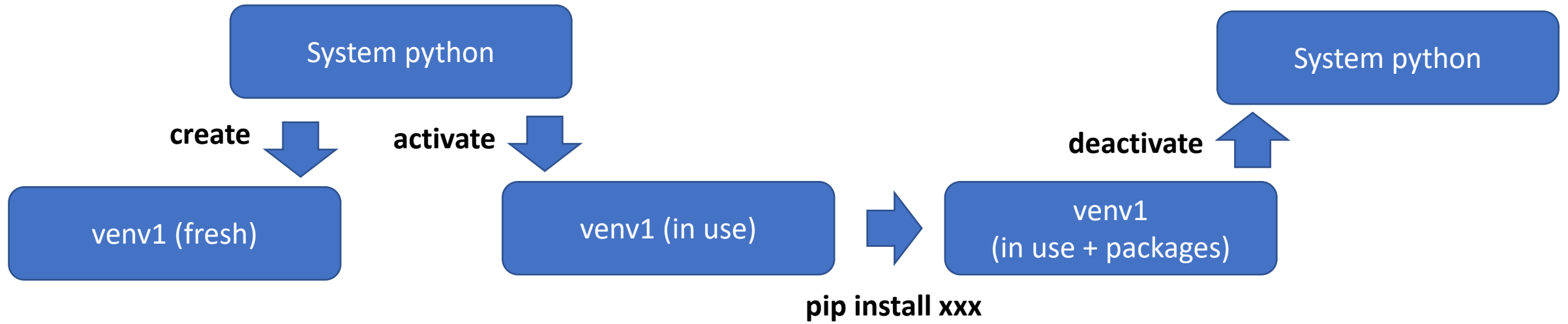
Virtual environments pattern



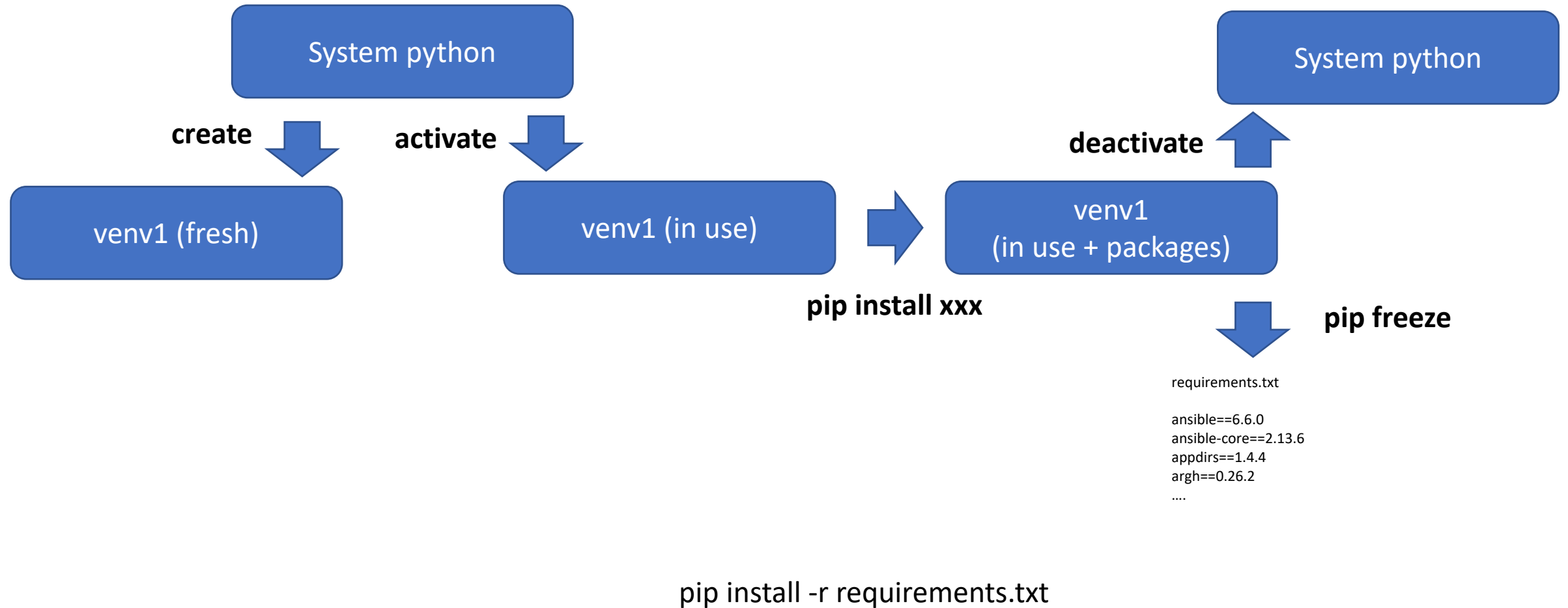
Virtual environments pattern



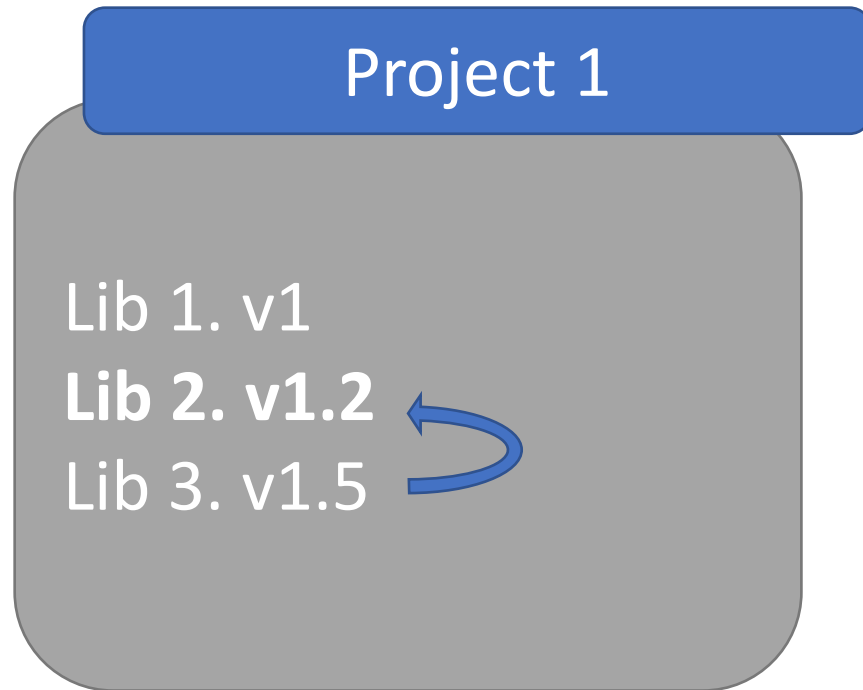
Virtual environments pattern



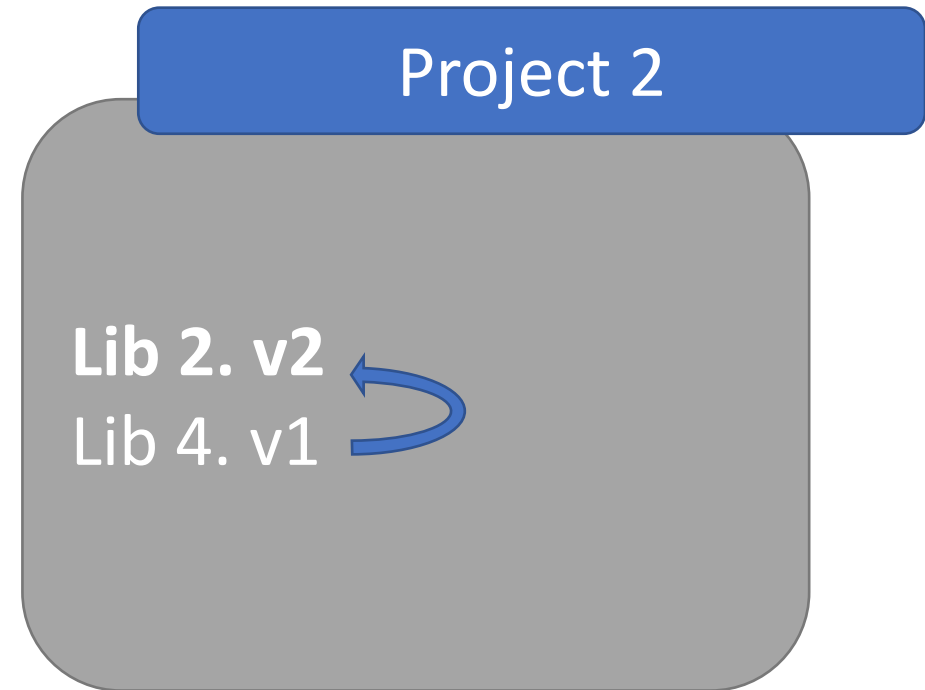
Virtual environments pattern



Python virtual environments



VENV1



VENV2

Basic implementation: **venv**

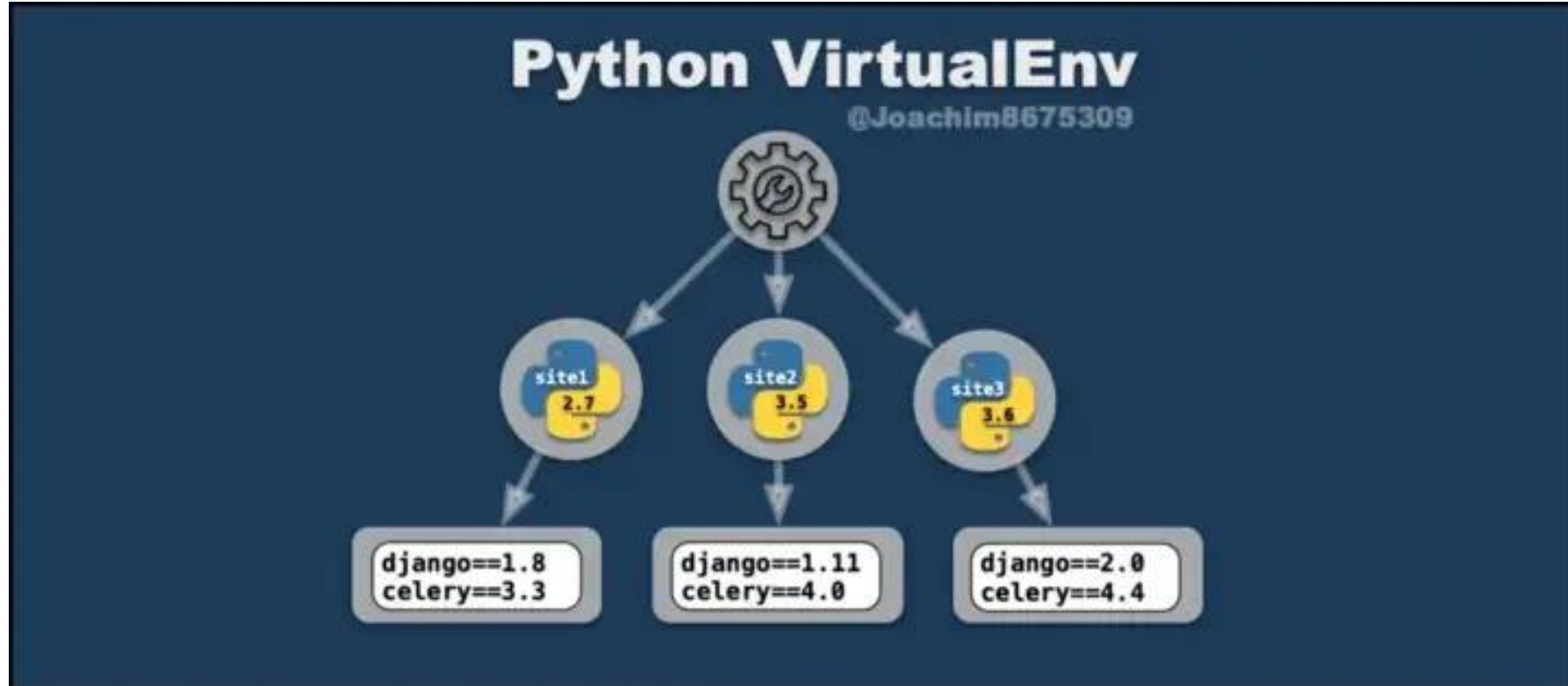
- **python3 -m venv env1**
 - **source env1/bin/activate**
- **pip** installs in dedicated path
- Dump all the packages:
 - `pip freeze > requirements.txt`



- Major limitation: same python version of the system

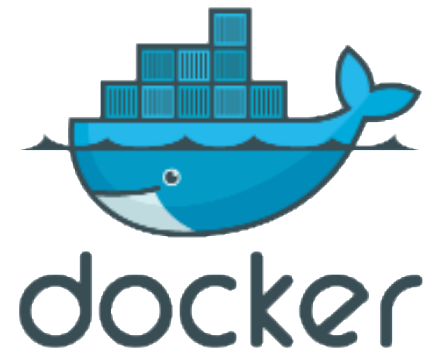
Python-level reproducibility: powerful alternatives

- **Virtualenv** or **conda environments**

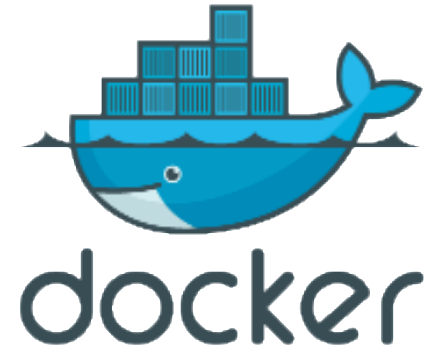


System-level reproducibility: containers

- Sandboxes
 - Run code in an “independent system” (nested)
 - Full config from “fresh” in **one file** (`Dockerfile`)
 - No scope on filesystem, ports, resources, unless specified binding
 - Full root rights

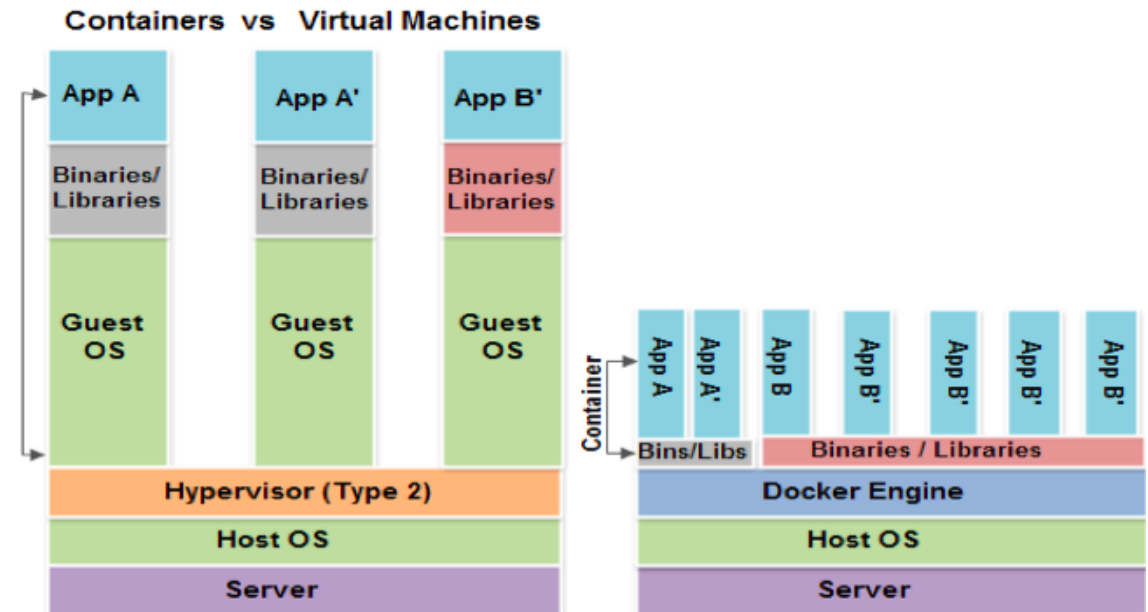


System-level reproducibility: containers



- Sandboxes
 - Run code in an “independent system” (nested)
 - Full config from “fresh” in **one file** (Dockerfile)
 - No scope on filesystem, ports, resources, unless specified binding
 - Full root rights

- Lighter than a VM
 - No nested kernel
 - Full definition in few bytes
 - Runs also on Win



Docker container: “fresh” system images

The screenshot shows the Docker Hub interface. At the top, there's a navigation bar with the Docker logo, a search bar containing 'python', and links for 'Explore', 'Pricing', 'Sign In', and 'Sign Up'. Below the navigation bar, there are tabs for 'Docker EE', 'Docker CE', 'Containers' (which is selected), and 'Plugins'. The main content area displays search results for 'python'. It shows '1 - 25 of 37,963 results for python. Clear search'. A dropdown menu is set to 'Most Popular'. The first result is 'python', marked as an 'OFFICIAL IMAGE'. It has '10M+' downloads and '4.1K' stars, and was 'Updated an hour ago'. The description says 'Python is an interpreted, interactive, object-oriented, open-source program...'. Below the description are tags for 'Container', 'Linux', 'Windows', '386', 'PowerPC 64 LE', 'IBM Z', 'x86-64', 'ARM', 'ARM 64', and 'Programming Languages'. The second result is 'pypy', also marked as an 'OFFICIAL IMAGE'. It has '10M+' downloads and '182' stars, and was 'Updated an hour ago'. The description says 'PyPy is a fast, compliant alternative implementation of the Python language.'. Below the description are tags for 'Container', 'Linux', '386', 'IBM Z', 'PowerPC 64 LE', 'ARM', 'x86-64', and 'Programming Languages'. On the left side, there are filters for 'Docker Certified' (with a checkbox and a blue checkmark), 'Images' (with checkboxes for 'Verified Publisher' and 'Official Images'), and 'Categories' (with checkboxes for 'Analytics', 'Application Frameworks', and 'Application').

<https://hub.docker.com>

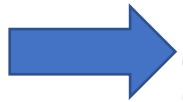
Docker container: pull & run

Pulling official image for python 3.7

```
docker run -i --rm -t python:3.7 /bin/bash
```

```
acorbe@nb-20-145 2019-collaborative-computing-ictp [master] $ docker run -i -t python:3.7 /bin/bash
Unable to find image 'python:3.7' locally
3.7: Pulling from library/python
e79bb959ec00: Extracting [=====> ] 44.5MB/45.34MB
d4b7902036fe: Download complete
1b2a72d4e030: Download complete
d54db43011fd: Download complete
69d473365bb3: Downloading [=====> ] 136.3MB/215MB
7dc3a6a0e509: Download complete
68cd774d0852: Download complete
2ef86095a118: Download complete
bd9da5a171e0: Download complete
```

Automated
download



```
[root@559da25be6b2:/# python --version
Python 3.7.3
```

Docker container: pull & run

Pulling official image for python 3.7

```
docker run -i -t python:3.7 /bin/bash
```

```
[root@559da25be6b2:/# python --version  
Python 3.7.3
```

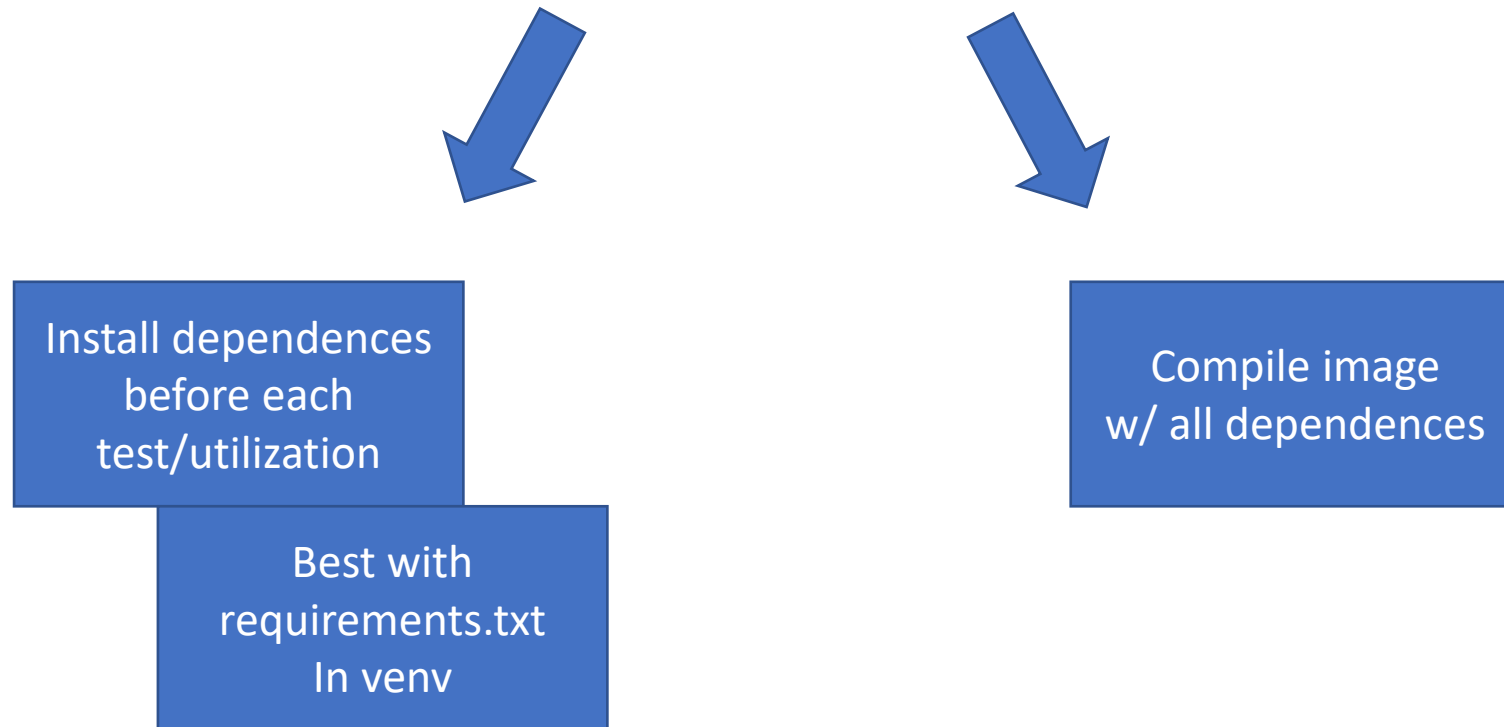
```
[root@559da25be6b2:/# cat /etc/os-release  
PRETTY_NAME="Debian GNU/Linux 9 (stretch)"  
NAME="Debian GNU/Linux"  
VERSION_ID="9"  
VERSION="9 (stretch)"  
ID=debian  
HOME_URL="https://www.debian.org/"  
SUPPORT_URL="https://www.debian.org/support"  
BUG_REPORT_URL="https://bugs.debian.org/"
```

We have a fresh Debian to run on!

Containers: fully reproducible environments

Ideal for “3rd” parties testing

What about dependencies? Libraries, etc.?



Case 1

Reconfig at every usage w/ script



image: python:3.8

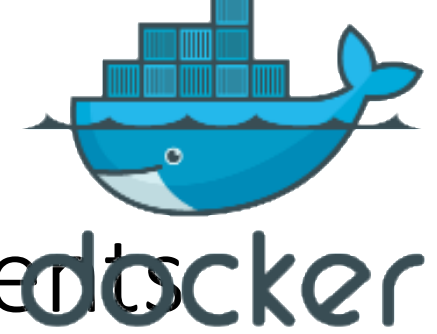
before_script:

- python -V # Print out python version for debugging
- pip install virtualenv
- virtualenv venv
- source venv/bin/activate
- pip install numpy nose
- # OR
- pip install -r requirements.txt

test:

script:

- cd binary_str_2_float
- nosetests -v



Case 2

Containers: fully reproducible environments
dependencies embedded (Dockerfile)

Similar to Makefile -> Dockerfile

```
docker build .
```

```
docker build -t tag:version .
```

Containers: fully reproducible environments

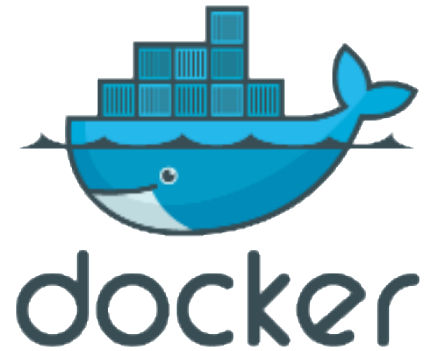
Where to start



Dockerfile 711 Bytes

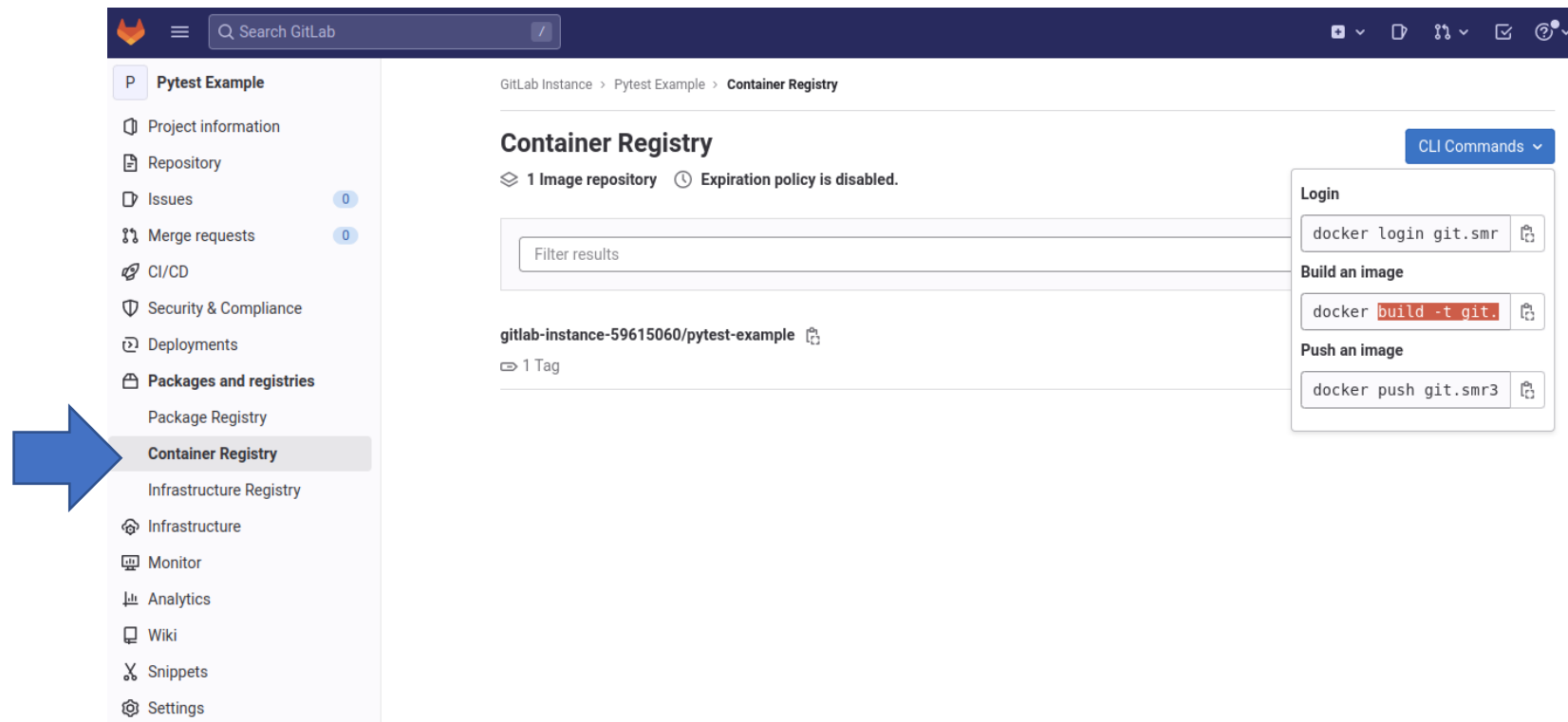
```
1 FROM ubuntu:18.04
2
3
4 # Install.
5 RUN \
6     sed -i 's/# \(..*multiverse$\)/\1/g' /etc/apt/sources.list && \
7     apt-get update && \
8     apt-get -y upgrade && \
9     apt-get install -y build-essential && \
10    apt-get install -y software-properties-common && \
11    apt-get install -y byobu curl git htop man unzip vim wget
12
13 ENV DEBIAN_FRONTEND=noninteractive
14
15
16 RUN \
17     apt-get install -y texlive-full
18
19 RUN \
20     apt-get install -y gnuplot
21
22 RUN \
23     rm -rf /var/lib/apt/lists/*
```

What to add



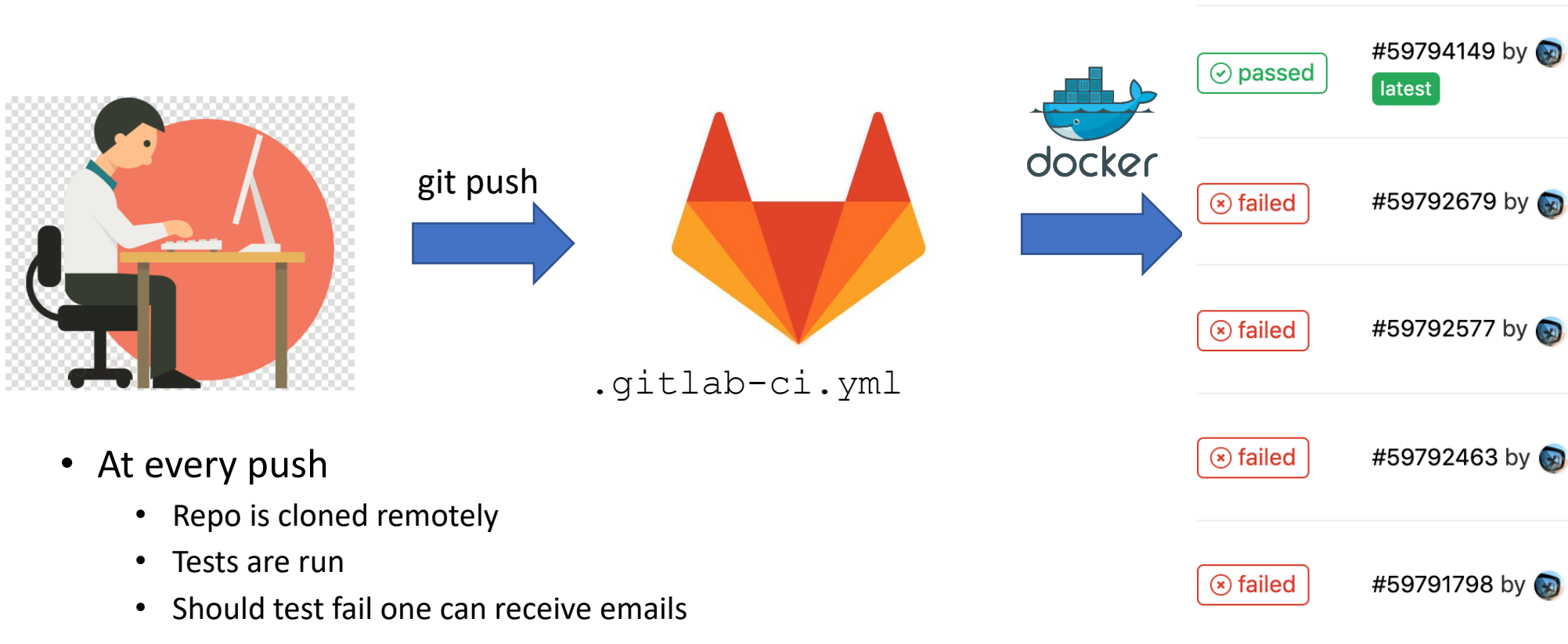
Gitlab container registry

- Built images can be pushed/pulled from the gitlab container registry



Some CI examples

Usecase 1: CI for testing - nose



- At every push
 - Repo is cloned remotely
 - Tests are run
 - Should test fail one can receive emails

CI for testing - pytest

image: python:3.6

variables:

PIP_CACHE_DIR: "\$CI_PROJECT_DIR/.cache"

cache:

paths:

- .cache/pip
- venv/

before_script:

- python -V # Print out python version for debugging
- pip install virtualenv
- virtualenv venv
- source venv/bin/activate
- pip install numpy pytest

test:

script:

- cd binary_str_2_float
- pytest -v

.gitlab-ci.yml

CI for testing - nose

 passed	#59794149 by 	 master  e391face  fixed converter		 00:01:08  2 minutes ago
 failed	#59792679 by 	 master  ab342d87  reduce		 00:01:36  23 minutes ago
 failed	#59792577 by 	 master  7284bc04  fixed print		 00:01:08  26 minutes ago
 failed	#59792463 by 	 master  c42b89c3  reverted to normal pyt...		 00:01:06  29 minutes ago
 failed	#59791798 by 	 master  854dd228  update		 00:00:35  41 minutes ago

CI for testing – pytest (here nosetest)

```
Successfully installed nose 1.3.7 numpy 1.10.3
$ cd binary_str_2_float
$ nosetests -v
tests that the trivial case 0.1e1 == 1. ... ok
tests that the trivial case 0.1111010101010e-4 == 0.0598907470703 up to 1e-10 accuracy ... ok
tests for the correct interpretation of negative sign ... ok
tests for the correct interpretation of positive sign ... ok
tests for the correct interpretation of negative exponent ... ok
tests for the correct interpretation of positive exponent ... ok
checks whether line command responds properly ... ok
Test that we deal only with normalized representations ... ok
Test that we deal only with normalized representations ... ok

-----
Ran 9 tests in 0.075s

OK
Creating cache default...
WARNING: .cache/pip: no matching files
venv/: found 2065 matching files
Uploading cache.zip to https://storage.googleapis.com/gitlab-com-runners-
cache/project/12175211/default
Created cache
Job succeeded
```

Duration: 1 m

Timeout: 1h (

Runner: share
5.gitlab.com (;

Commit [e391](#)

fixed converte

✓ Pipeline #!

test

➔ ⏸ test

Usecase 2: Building binaries -- latex

- Version control on Latex:
 - Stop with `paper_0.tex`; `paper_1`; `paper_2.tex`; `paper_final.tex`; `paper_edits.tex`...
 - Simple collaboration with co-authors
 - Simply tracing source
- Issue: we want our paper to build
 - now and in 10 years; regardless the env
 - i.e. all dependencies are present
 - Let's make it a CI-job

Building binaries -- latex

```
.gitlab-ci.yml 289 Bytes
1
2
3  job:compile:
4    image: niccokunzmann/ci-latex
5    script:
6      - latexmk -pdf
7      - echo 'renaming target file for artifact upload...'
8      - mv template.pdf paper-${CI_COMMIT_SHORT_SHA}.pdf
9    artifacts:
10     paths:
11       - paper-${CI_COMMIT_SHORT_SHA}.pdf
12
13
14
15
16
17
```

- <https://gitlab.com/acorbe/ci-latex-example>

Building binaries -- latex



job:compile [Retry](#)

Duration: 4 minutes 25 seconds
Timeout: 1h (from project) [?](#)
Runner: shared-runners-manager-5.gitlab.com (#380986)

Job artifacts

[Download](#) [Browse](#)

Commit [3cac8253](#) [🔗](#)
Update template.tex

✓ **Pipeline** [#59102319](#) for [master](#)

- At every push
 - Paper is built – code/dependency check
 - Automating spell check would be possible
 - Served as CI product + hash tracking

✓ **passed** **Job #204853357** in pipeline **#59102319** for **3cac8253** from **mas**

 **Alessandro Corbetta** 4 days ago

Artifacts [Download artifacts](#)

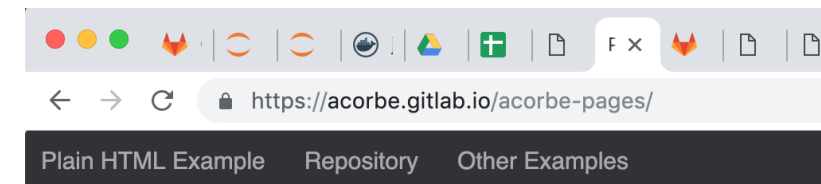
Name	Size
 paper-3cac8253.pdf	165 KB

Usecase 3: CI static websites

<https://gitlab.com/acorbe/acorbe-pages>

Name
public
 .gitlab-ci.yml
 README.md

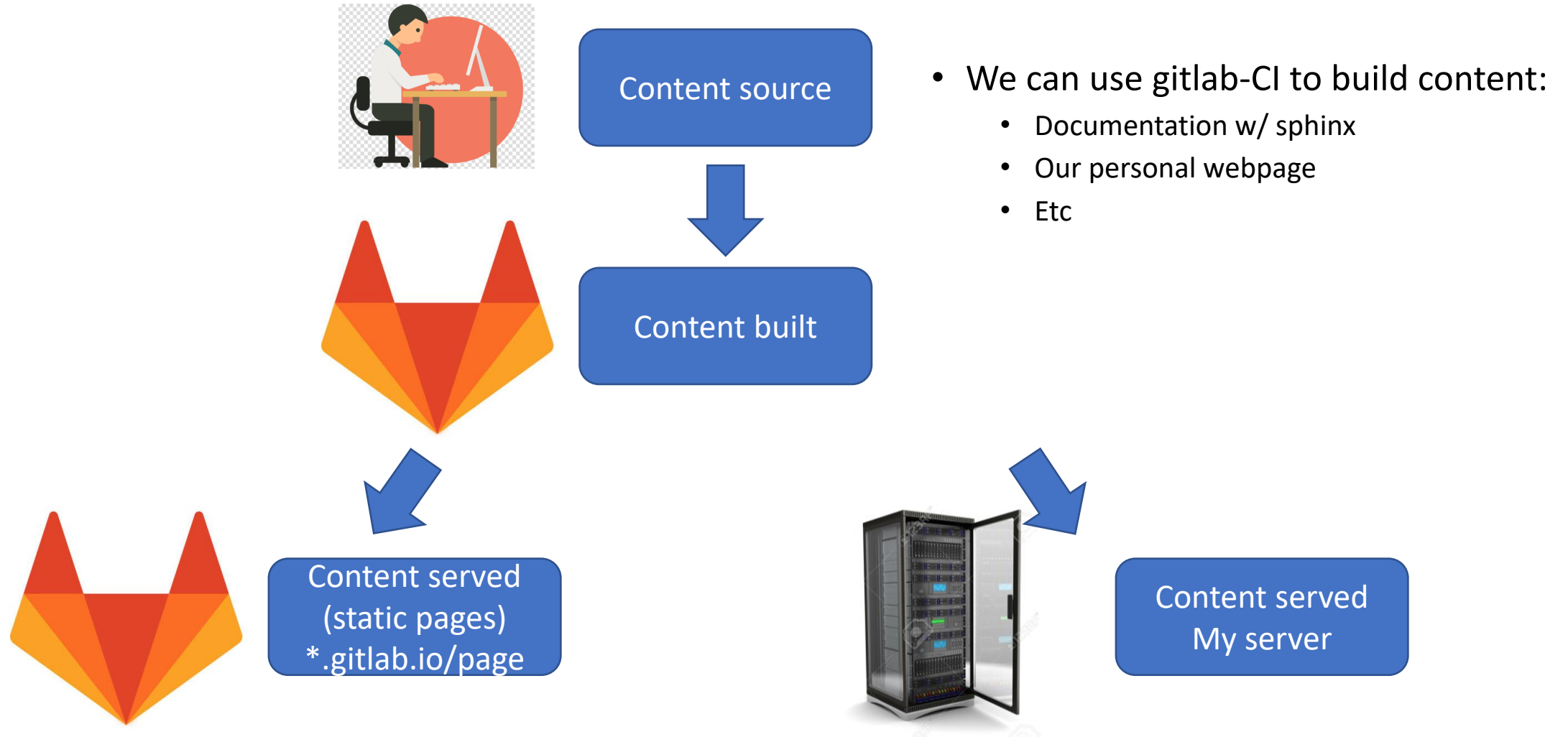
```
 .gitlab-ci.yml 139 Bytes   
1  image: alpine:latest  
2  
3  pages:  
4    stage: deploy  
5    script:  
6      - echo 'Nothing to do...'  
7  artifacts:  
8    paths:  
9      - public  
10  only:  
11    - master
```



Hello World!

Example -- This is a simple plain-HTML website on GitLab Pages, with

Usecase 4: CI for deploying content pages/documentation



Assumption: sphynx config in /docs

 .gitlab-ci.yml  516 bytes

Edit in pipeline editor ▼ Replace Delete  

```
1 image: python:3.7
2
3
4 stages:
5   - docs
6
7
8 pages:
9   stage: docs
10
11   script:
12     - apt-get update
13     - apt-get install -y graphviz
14     - pip3 install numpy pandas scipy networkx matplotlib sphinx sphinxcontrib-napoleon pydot dask toolz pyreadr sklearn numba
15     - pip3 install dask[dataframe] --upgrade
16     - rm -f /usr/bin/python && ln -s /usr/bin/python3 /usr/bin/python
17     - cd docs
18     - make html
19     - mv build/html ../public
20   artifacts:
21     paths:
22     - public
```

Usecase 5: deployment on external server

stages:

- generate_pdfs
- build_site

generate_my_resume:

stage: generate_pdfs

script:

- apt update
- apt install -y python-pip - pip install jinja2
- make my_resume
- cp resume/resume-corbetta.pdf content/publ

artifacts:

paths:

- content/publ/resume-corbetta.pdf

build-and-deploy:

stage: build_site

environment:

name: corbetta-website

url: http://corbetta.phys.tue.nl

script:

- source activate pelican
- make ssh_upload

Pipeline Jobs 2

Generate_pdfs

Build_site



generate_my_re...



build-and-depl...

