Buildpacks for reproducibility

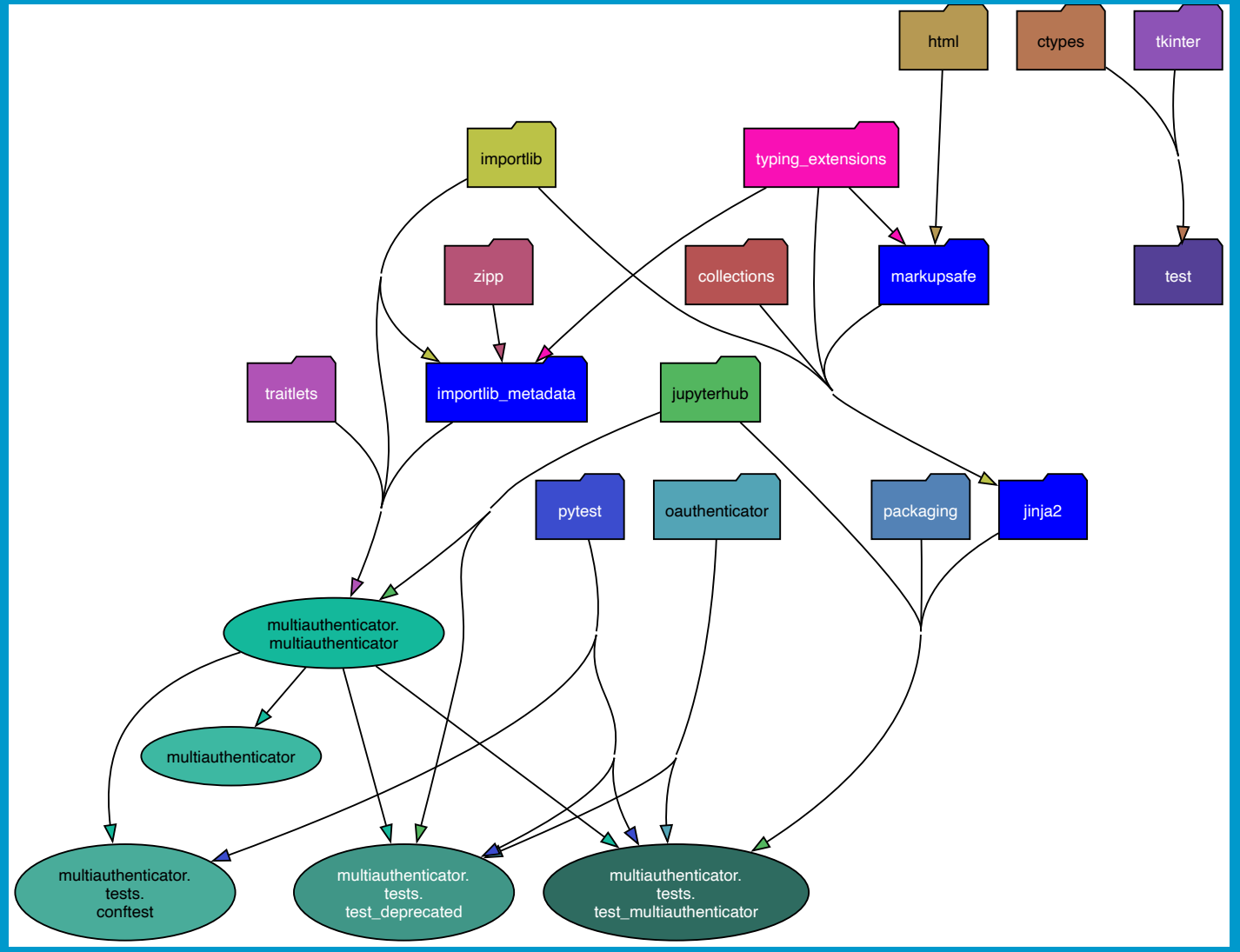
Samuel Gaist (Idiap Research Institute)

EnhanceR Symposium 2025

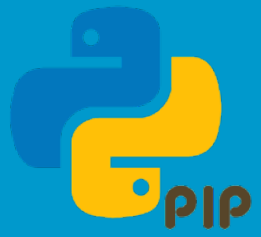
What is reproducibility ?

- Known dependencies
- Known environment
- Known data

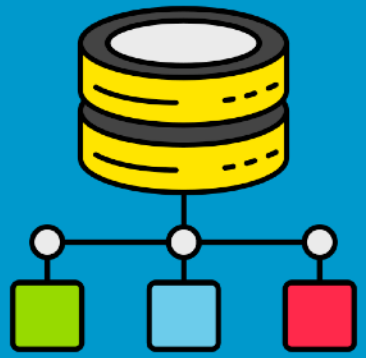
Known dependencies

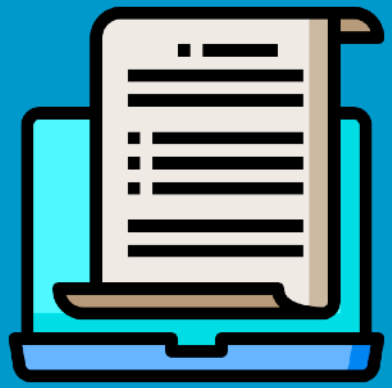


Known environment



Known data



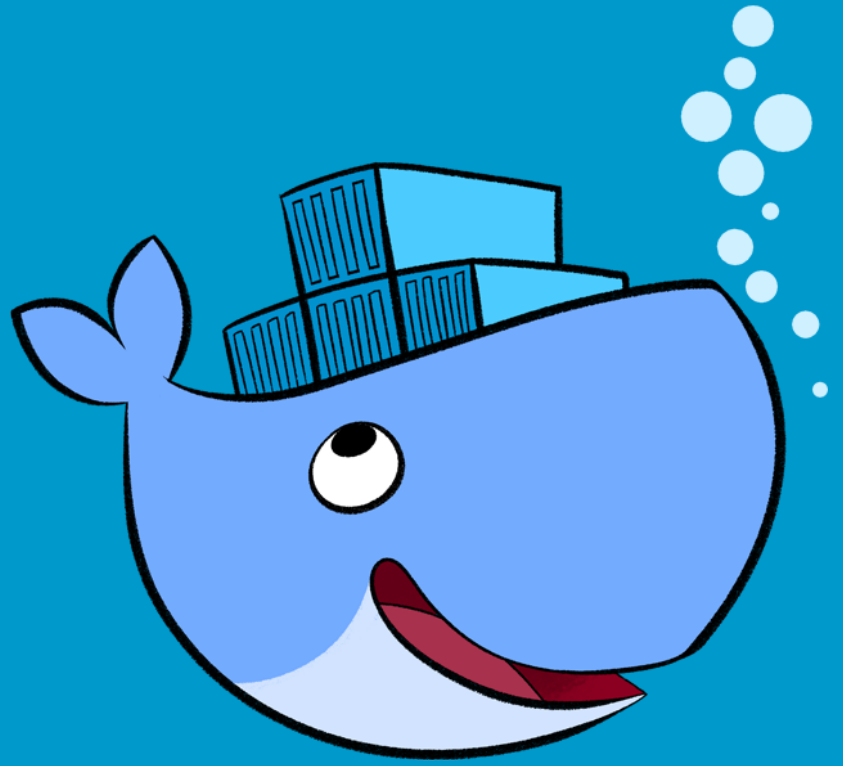


How to do reproducibility ?

- It works on my machine ! -> No
- Copy that 150GB folder ! -> No
- Take my code and use it ! -> No
- Have this Docker image -> Now we are getting somewhere

Lets build a Docker image

- Learn the Dockerfile format
- Learn the Docker pitfalls
- Learn how to use .dockerignore
- Learn how to not build an overly large image



Can that be made simpler ?

- Buildpacks

- Paketo’s pack command

What are buildpacks ?

- Started with Heroku
- Now CNB (Cloud Native Buildpacks)
- Set of scripts and configuration files to turn code into a deployable artifact with minimal configuration
- Allows to concentrate on code
- Allows for reproducible image build
- No Dockerfile involved

How do buildpacks work ?

- Language for the buildpacks:
  - Does not matter as long as it’s executable
- Paketo:
  - Set of buildpacks for popular languages and frameworks
  - Production ready
  - Easy to integrate in build pipelines
  - Written in Go

How do buildpacks work ?

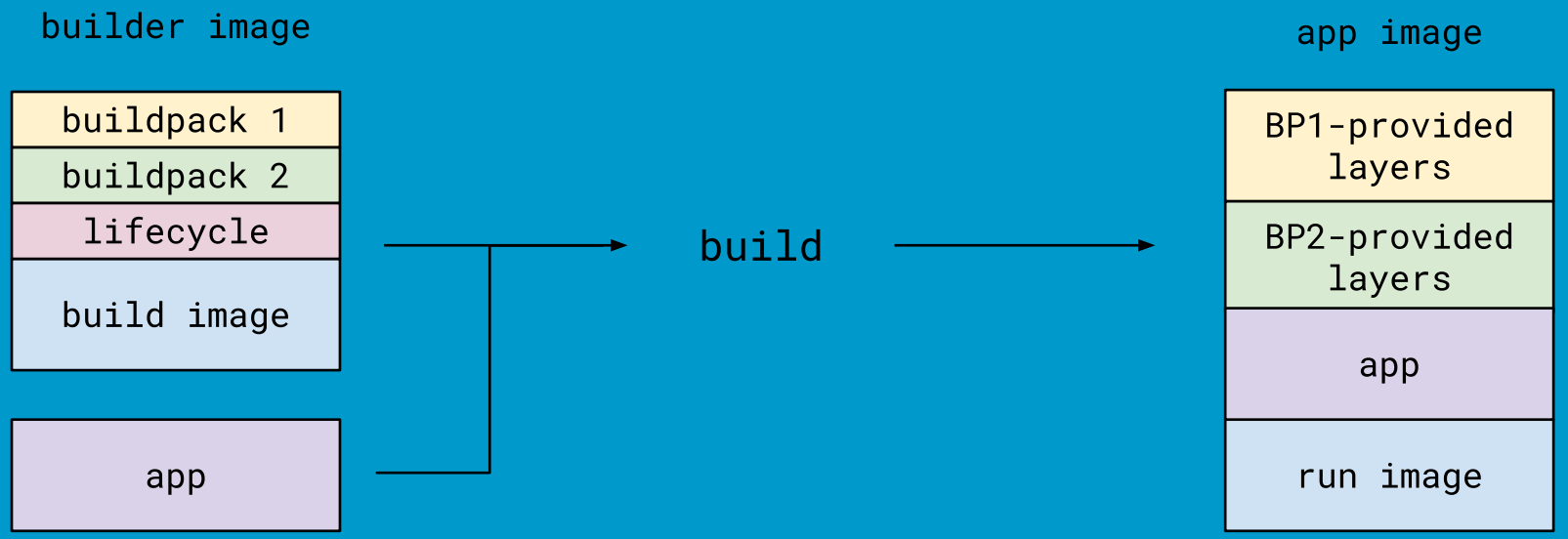
- Detection
  - Search for known language files (requirement.txt, package.json, etc.)
  - Search for known configuration file (project.toml, runtime.txt, Procfile)
- Build
  - Install language specific tools
  - Setup / install project
  - Project specific configuration (e.g. run time version, env var, etc.)

How do buildpacks work ?

- Layers:
  - Build
  - Launch
  - Cache
- Each layer is a self-contained folder
- The base layer can be swapped without having to rebuild everything
- Cache is local to the build machine

How do buildpacks work ?

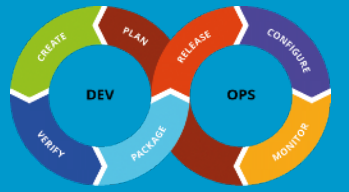
What happens at build time



How to use pack ?

- CLI

- CI/CD pipelines



- Kubernetes (kpack, Shipwright)

How to use them ? (cli)

```
pack build test_img --path apps/test-app --builder cnbs/sample-builder:bionic
```

How to use them ? (CI/CD)

```
pack build test_img --path apps/test-app --builder cnbs/sample-builder:bionic
```

How to use them ? (kpack)

```yaml
apiVersion: kpack.io/v1alpha2
kind: Build
metadata:
  name: sample-build # This can be any name
spec:
  tags:
    - <app-image-name>
  builder:
    image: cnbs/sample-builder:<bionic OR alpine>
  source:
    git:
      url: https://github.com/buildpacks/samples.git
      revision: main
    subPath: "apps/<APP>"
```

How to use them ? (shipwright)

- BuildStrategy: Buildpacks v3
- Build: Settings for building a project
- BuildRun: Actual build

What do you get ?

- Your project only contains your code / configuration
- An image ready to be used

Where is it used ?

- GitLab Auto DevOps pipeline
- The Renku research platform uses it for custom sessions

Key Takeways

- Creating reproducible environments is easy
- Buildpacks are convenient building blocks
- You can concentrate on your code

Questions ?

Thank you for your attention !

Contact: [samuel.gaist@idiap.ch]

Credits

Dependency graph, cli, yaml example: Idiap Research Institute

conda icon: https://www.anaconda.com/

uv icon: https://astral.sh

Poetry icon: https://python-poetry.org

Data Icons: https://www.flaticon.com

Docker image: https://cleanpng.com

Cloud Native Buildpacks (Logo and build explanation): https://buildpacks.io

Paketo: https://paketo.io

Icons not listed above: https://cleanpng.com