UFRN
UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

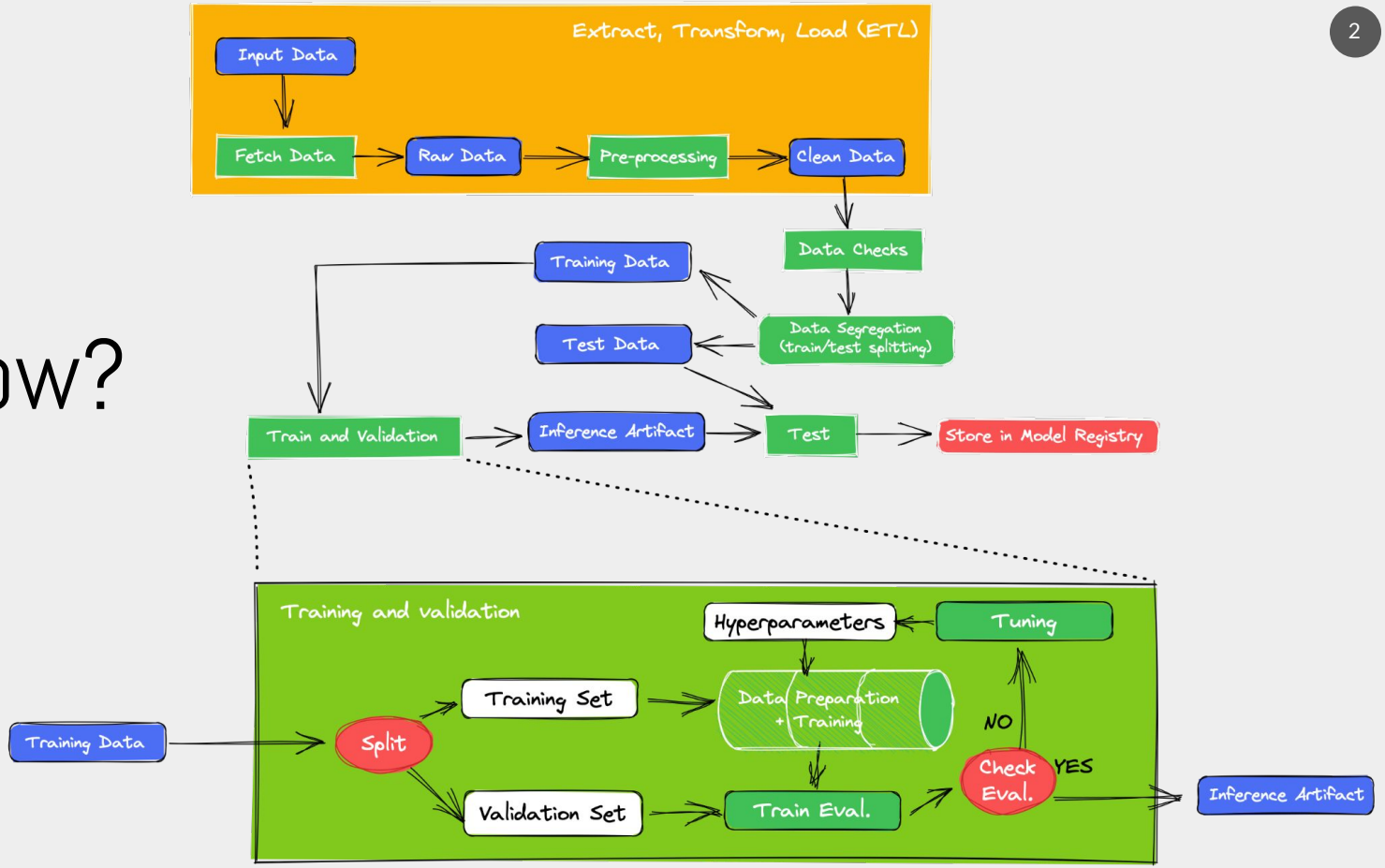
NE 4.0

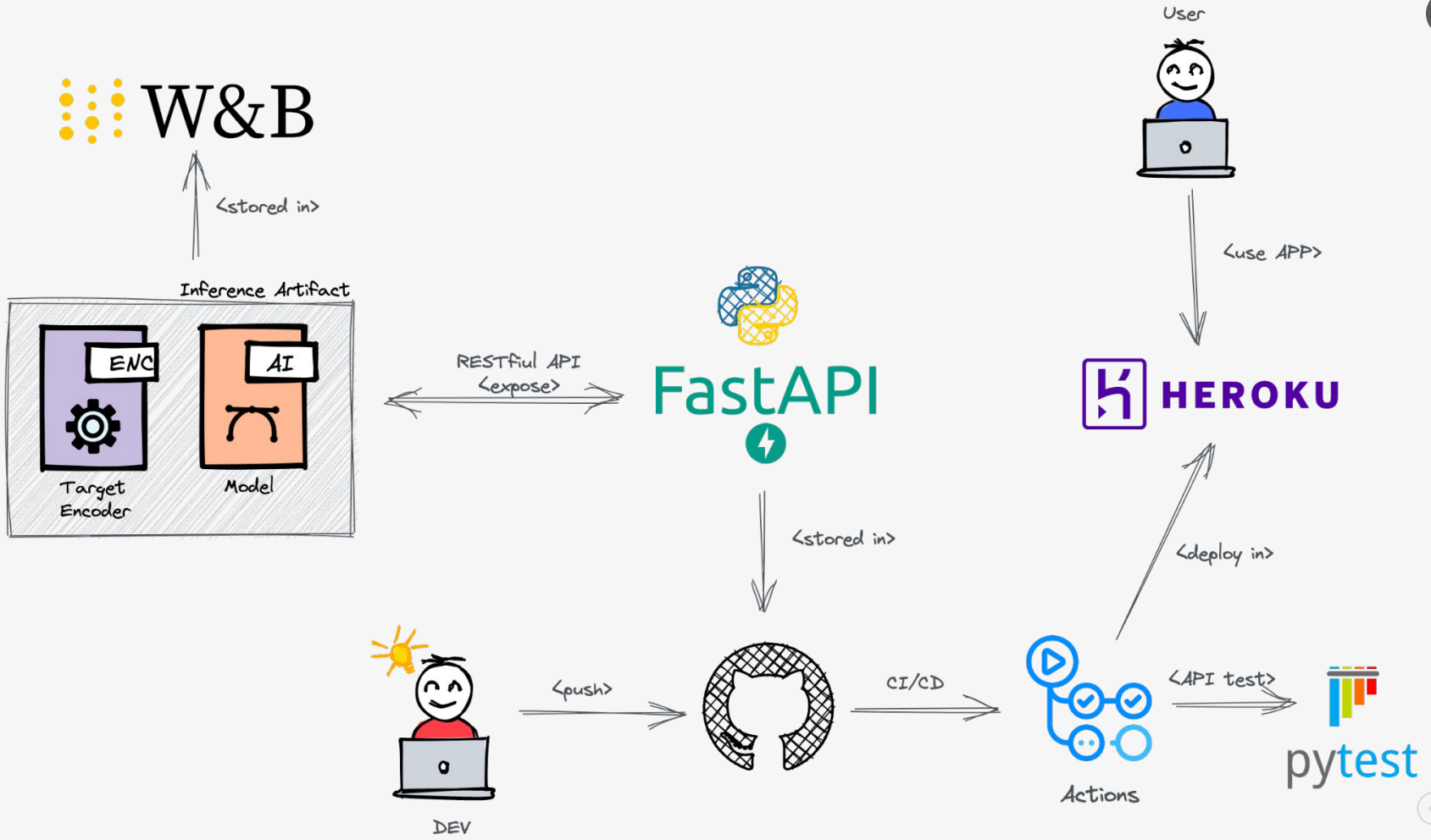
Aprendizado de Máquina

Implantando um modelo de AM em produção

Ivanovitch Silva
ivanovitch.silva@ufrn.br

What now?


Extract, Transform, Load (ETL)
Input Data
Fetch Data
Raw Data
Pre-processing
Clean Data
Data Checks
Training Data
Data Segregation (train/test splitting)
Test Data
Train and Validation
Inference Artifact
Test
Store in Model Registry
Training and validation
Hyperparameters
Tuning
Training Data
Split
Training Set
Data Preparation + Training
NO
Check Eval.
YES
Validation Set
Train Eval.
Inference Artifact


User
W&B
<stored in>
<use APP>
Inference Artifact
ENC
AI
Target Encoder
Model
RESTful API
<expose>
FastAPI
HEROKU
<stored in>
<deploy in>
<push>
CI/CD
<API test>
pytest
Actions
DEV

Step #01

Data science technology for a better world.

Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today.

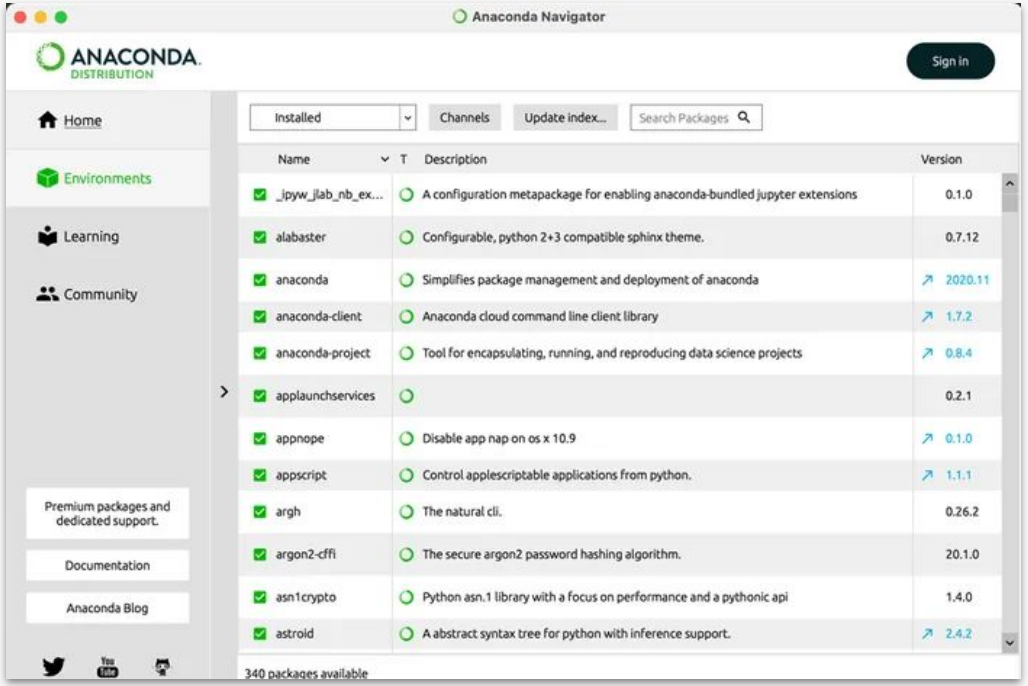
Download

For MacOS

Python 3.9 • 64-Bit Graphical Installer • 515 MB

Get Additional Installers

jupyter CONDA®



https://www.anaconda.com

Step #02

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * **Repository name** *

ivanovitchm /

Great repository names are short and memorable. Need inspiration? How about animated-invention?

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.

.gitignore template: Python

GitHub

Step #03

▶️ Environment Setup

Create a conda environment with `environment.yml`:

```
conda env create --file environment.yml
```

To remove an environment in your terminal window run:

```
conda remove --name myenv --all
```

To list all available environments run:

```
conda env list
```

To activate the environment, use

```
conda activate myenv
```

```yaml
name: colab2deploy
channels:
  - conda-forge
  - defaults
dependencies:
  - numpy=1.21.5
  - uvicorn=0.17.5
  - gunicorn=20.1.0
  - requests=2.27.1
  - fastapi=0.74.0
  - scikit-learn=1.0.2
  - python=3.8
  - jupyterlab=3.2.9
  - jupyter=1.0.0
  - ipywidgets=7.6.5
  - jupyterlab_widgets=1.0.2
  - git=2.34.1
  - pydantic=1.9.0
  - yaml=0.2.5
  - pip=21.3.1
  - pandas=1.3.5
  - pytest=6.2.5
  - wandb=0.12.14
```

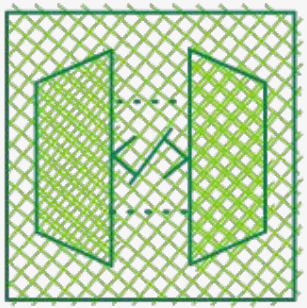
Step #04

An ASGI web server, for Python.

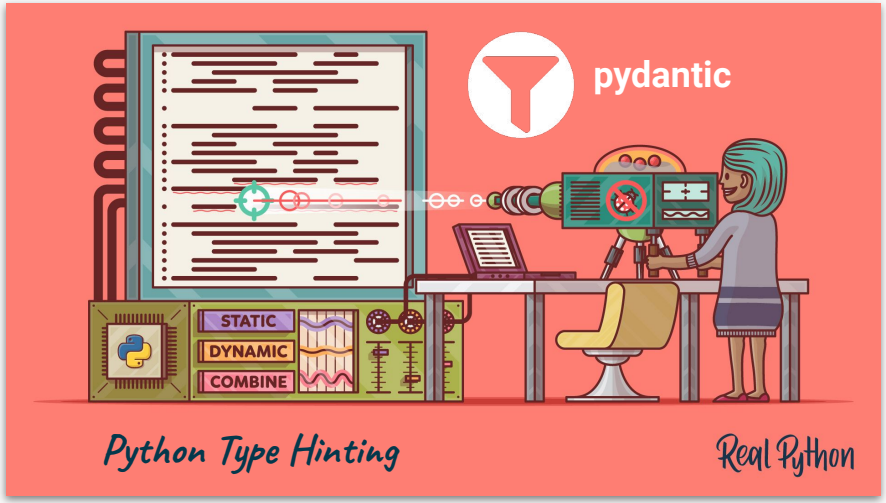
FastAPI

```python
from typing import Union

def foo(a: Union[list,str], b: int = 5) -> str:
    pass
```



1) Web framework for developing RESTful APIs in Python
2) It is fast in execution and also in development
3) Easy to use (CRUD) and production-friendly
4) post, get, put, patch, delete docs, path, query, authentication



Step #05 Local API Testing

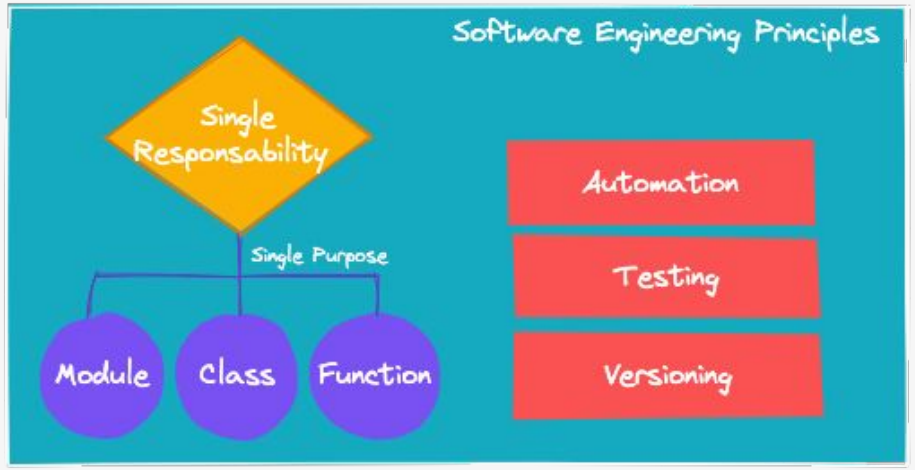

```
from fastapi.testclient import TestClient
```

Runing FastAPI via uvicorn
is easy, but clunky for testing

FastAPI

```
(colab2deploy) ❯ uvicorn source.query.main:app --reload                    colab2mlops -> main !
INFO:     Will watch for changes in these directories: ['/Users/ivanovitchsilva/colab2mlops']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [96831] using statreload
INFO:     Started server process [96833]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     127.0.0.1:57620 - "GET / HTTP/1.1" 200 OK
```

Step #06

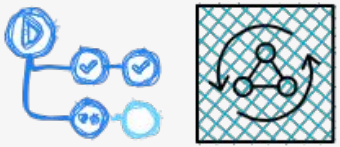


CI/CD

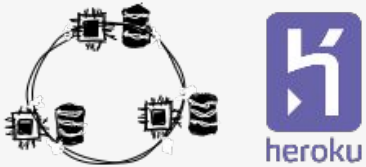
Core driver of putting Software engineering principles into practice

The practice of often fitting code into the overall code base

- Test suite
- Build/compile whenever we push changes



Continuous Integration (CI)

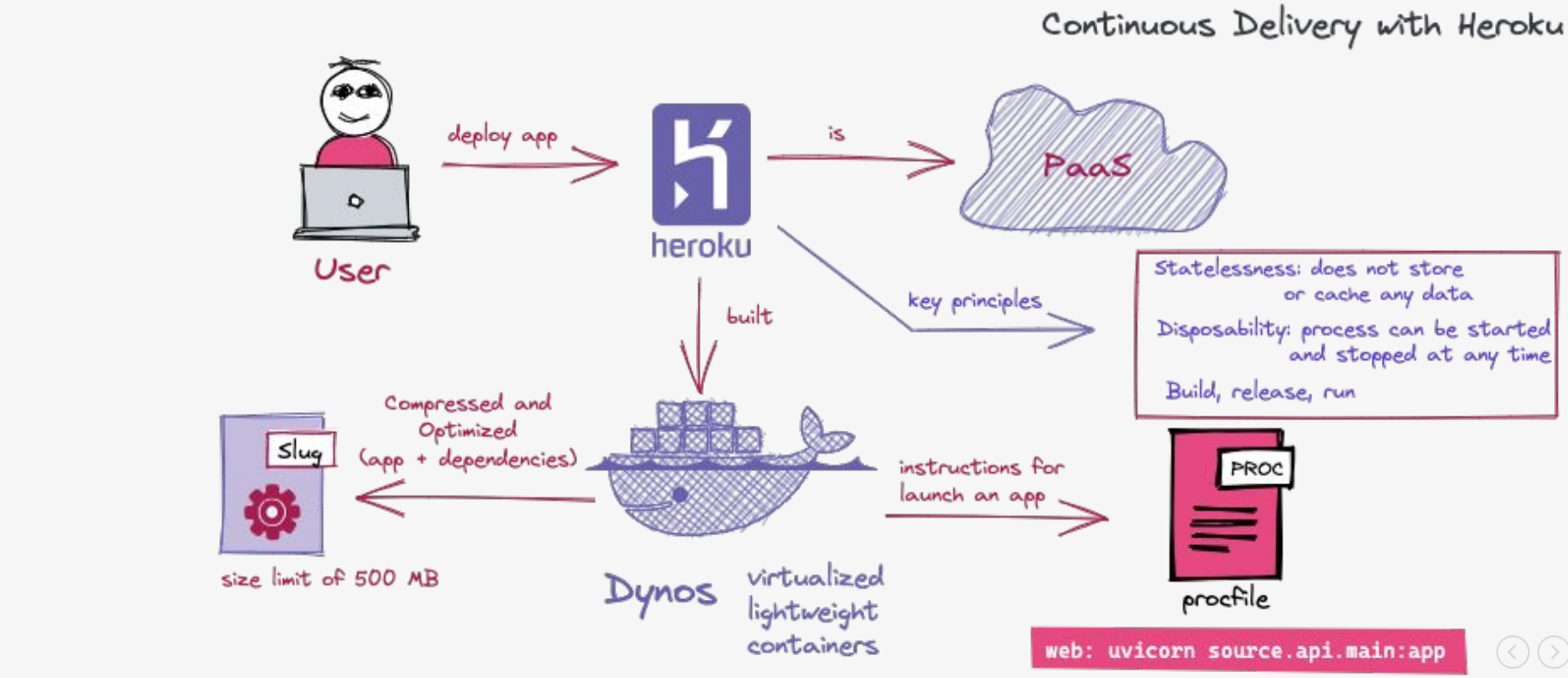


Continuous Delivery (CD)

The practice of making our code always deployed

- Code gets verified by CI then auto-pushed into production

Step #07


Continuous Delivery with Heroku
deploy app
User
heroku
is
PaaS
key principles
Statelessness: does not store or cache any data
Disposability: process can be started and stopped at any time
Build, release, run
built
Compressed and Optimized (app + dependencies)
Slug
size limit of 500 MB
Dynos
virtualized lightweight containers
instructions for launch an app
PROC
procfile
web: uvicorn source.api.main:app