#### Building Reproducible Pipelines

* [Building and Deploying a Reproducible Machine Learning Pipeline](https://trainindata.medium.com/how-to-build-and-deploy-a-reproducible-machine-learning-pipeline-20119c0ab941) - article
* [Building a Reproducible Machine Learning Pipeline](https://arxiv.org/ftp/arxiv/papers/1810/1810.04570.pdf) - long article
* [Reproducible Machine Learning](http://www.rctatman.com/files/Tatman_2018_ReproducibleML.pdf) - presentation, Kaggle
* [The Machine Learning Reproducibility Crisis](https://petewarden.com/2018/03/19/the-machine-learning-reproducibility-crisis/) - article, by Google developer

#### Streamlining Deployment with Open Source

* [Six motivations for using open source](https://opensource.com/life/15/12/why-open-source) – article

#### Further Reading Materials

Python Conventions

- [PEP8](https://www.python.org/dev/peps/pep-0008/)

- [PEP 484 - typehints](https://www.python.org/dev/peps/pep-0484/)

- [Using requirements files](https://realpython.com/lessons/using-requirement-files/)

- [Tox Overview](https://christophergs.com/python/2020/04/12/python-tox-why-use-it-and-tutorial/" \t "_blank)

- [Building Python Packages](https://packaging.python.org/tutorials/packaging-projects/)

- [Why not use Python for config](https://hitchdev.com/strictyaml/why-not/turing-complete-code/)

- [Primer on pyproject.toml](https://snarky.ca/what-the-heck-is-pyproject-toml/)

Tox -e run

Python app/main.py

**Lecture 1 (Demo)**

* [OpenAPI Specification](https://swagger.io/specification/)
* [curl documentation](https://curl.se/docs/)

**Lecture 2 (Architecture)**

* [Understanding REST APIs](https://www.smashingmagazine.com/2018/01/understanding-using-rest-api/)
* What is [AJAX](https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX/Getting_Started)
* [ReactJS documentation](https://reactjs.org/docs/getting-started.html)
* Martin Fowler on [Microservice pre-requisites](https://martinfowler.com/bliki/MicroservicePrerequisites.html)

**Lecture 3 (FastAPI Intro)**

* [FastAPI documentation](https://fastapi.tiangolo.com/)
* Python [Asycio documentation](https://docs.python.org/3/library/asyncio.html" \t "_blank)
* Python [type hints documentation](https://docs.python.org/3/library/typing.html)

**Lecture 4 (API Endpoints)**

* The [Python Package Index](https://pypi.org/)
* [Pydantic docs](https://pydantic-docs.helpmanual.io/)

**Lecture 5 (API Schemas)**

* [OpenAPI Specification](https://swagger.io/specification/)

**Lecture 6 (Logging)**

* Python [logging standard library documentation](https://docs.python.org/3/library/logging.html)
* [Loguru Intercept [advanced]](https://loguru.readthedocs.io/en/stable/overview.html#entirely-compatible-with-standard-logging)

**Lecture 7 (Uvicorn)**

* [Uvicorn documentation](https://www.uvicorn.org/)

**Lecture 8 (PaaS)**

* [Heroku Pricing](https://www.heroku.com/pricing)

**Lecture 9 (Deployment)**

* [Git subtree](https://www.atlassian.com/git/tutorials/git-subtree)

**Lecture 10 (Heroku files)**

* Heroku Procfile [docs](https://devcenter.heroku.com/articles/procfile)

Section Notes & Further Reading

* [Introduction to CI/CD](https://www.digitalocean.com/community/tutorials/an-introduction-to-continuous-integration-delivery-and-deployment)
* [CircleCI workflows](https://circleci.com/blog/introducing-workflows-on-circleci-2-0/)
* [Filter workflows by branch](https://support.circleci.com/hc/en-us/articles/115015953868-Filter-workflows-by-branch-) (the very observant students will notice how in lecture 3 the job still runs, even though I haven't merged it into master yet - I cheated.)
* [CircleCI use environment variables](https://circleci.com/docs/2.0/env-vars/)
* [Gemfury Tokens](https://gemfury.com/help/pypi-server/)
* [Github Forks](https://docs.github.com/en/get-started/quickstart/fork-a-repo#:~:text=Celebrate-,About%20forks,without%20affecting%20the%20upstream%20repository.)
* [Get Heroku API key](https://devcenter.heroku.com/articles/authentication)
* [The kaggle CLI](https://github.com/Kaggle/kaggle-api)
* [Git tags](https://git-scm.com/book/en/v2/Git-Basics-Tagging)

11.5a: Heroku-Docker Gotchas

#### Heroku with Docker Gotchas

If during your docker image release to Heroku you get: Expected response to be successful, got 404

Then simplify the Makefile jobs to not include the commit ID and just use "latest":

(note that Makefiles require tabs, not spaces)

1. NAME=udemy-course-dummy
2. COMMIT\_ID=$(shell git rev-parse HEAD) # REMOVED

5. build-ml-api-heroku:
6. docker build --build-arg PIP\_EXTRA\_INDEX\_URL=${PIP\_EXTRA\_INDEX\_URL} -t registry.heroku.com/$(NAME)/web .
8. push-ml-api-heroku:
9. docker push registry.heroku.com/${HEROKU\_APP\_NAME}/web:latest

Note that your Makefile variable NAME (not a good var name, obviously) should match your HEROKU\_APP\_NAME

This is not an issue we encountered when constructing the course, but is now present, which could be due to undocumented changes to the Heroku API. Best practice is to use a commit hash (or some specific version ID), and you can do that via the API as documented here: <https://devcenter.heroku.com/articles/container-registry-and-runtime#api>