

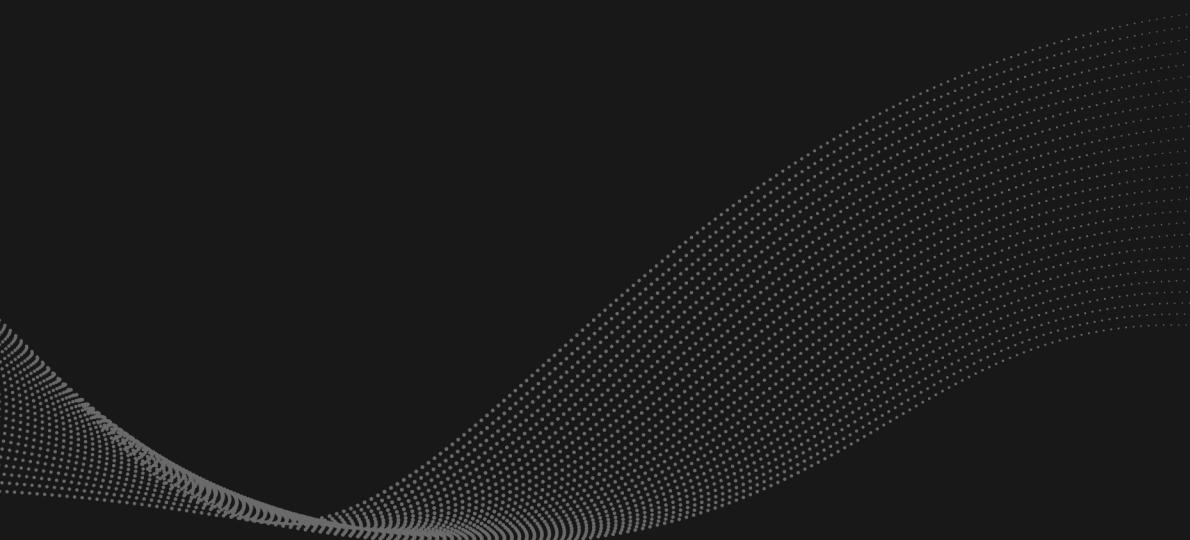


Lab 01: Introduction to MLOps

The MLOps Course

22 September 2025

Table of Contents

- 
- 01 Introduction
 - 02 Git
 - 03 Python environments
 - 04 Simple ML Pipeline
 - 05 Conclusion

Lab Structure

- follows the lectures
- practical exercises



presentations
high-level overview



hands-on exercises
explore, experiment,
learn by doing



bonus points
solve first + correctly



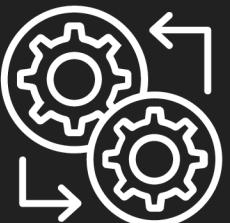
solution
walk-through, discussion,
shared on github

What is MLOps

- Machine Learning + DevOps
- deploy + maintain ML Models in prod



Version Control
track changes in
code, data, models



CI/CD Pipelines
automate integration,
testing, deployment

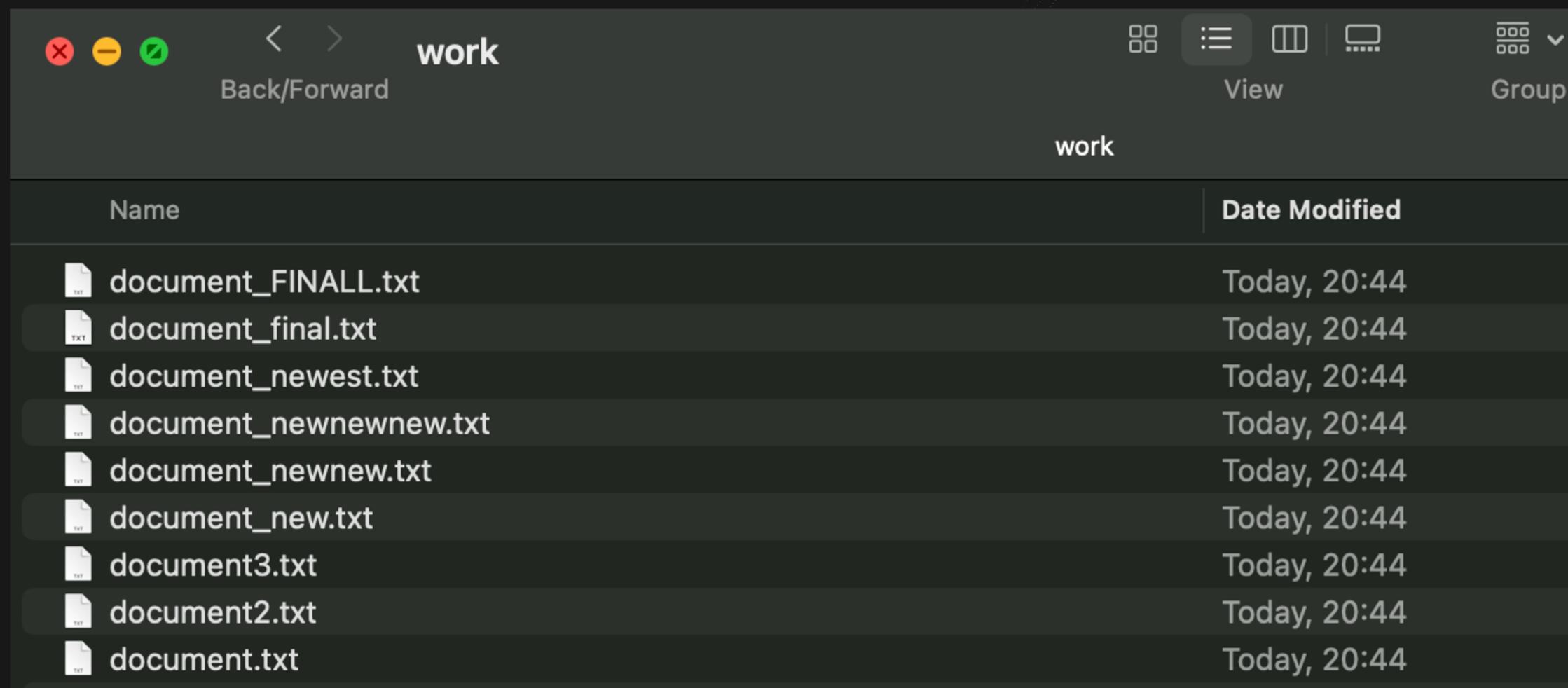


Monitoring and Logging
performance + logs



Automated Testing
performance
expectations

Basics of Version Control



The screenshot shows a file browser window titled "work". The toolbar includes icons for Back/Forward, View, and Group. The main area displays a list of files with the following data:

Name	Date Modified
document_FINAL.txt	Today, 20:44
document_final.txt	Today, 20:44
document_newest.txt	Today, 20:44
document_newnewnew.txt	Today, 20:44
document_newnew.txt	Today, 20:44
document_new.txt	Today, 20:44
document3.txt	Today, 20:44
document2.txt	Today, 20:44
document.txt	Today, 20:44

Basics of Version Control

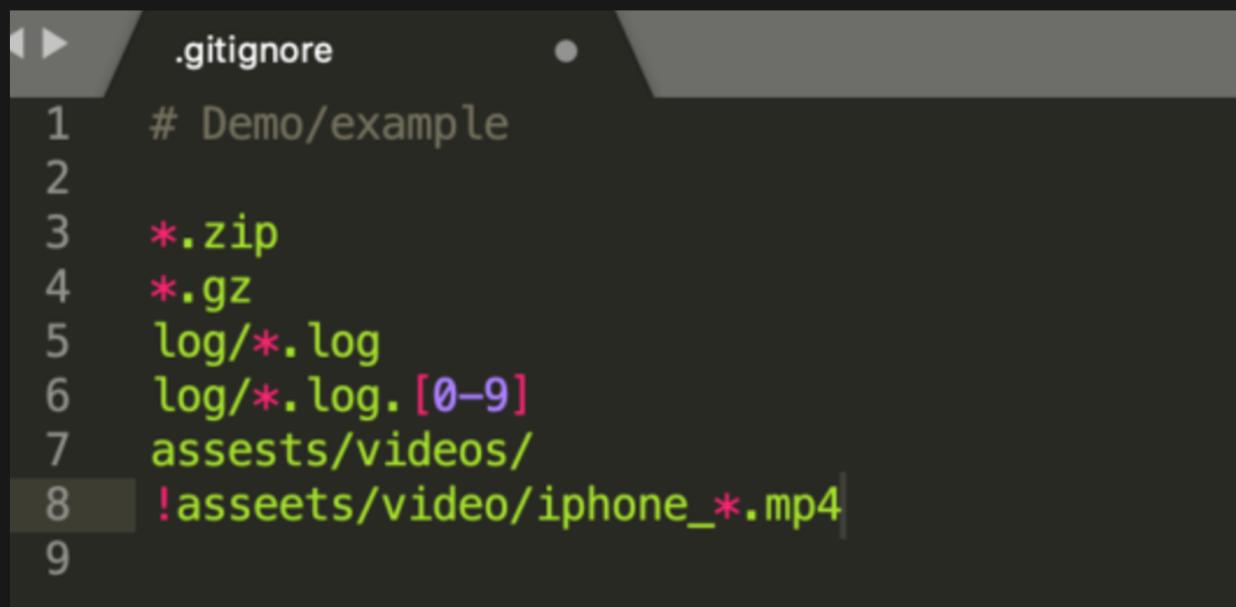
- system for tracking changes
- latest version + history tracker

Name	Last commit message	Last commit date
..		
document.txt	final touches and formatting	2 minutes ago

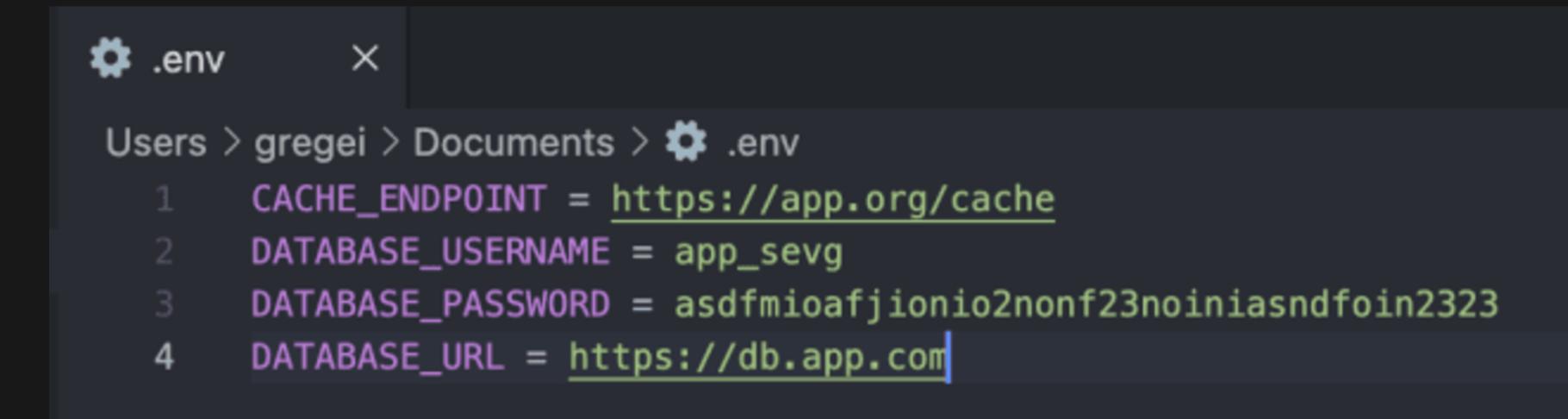
History for document.txt on main	
-o-	Commits on Sep 1, 2024
	final touches and formatting
 natalijamitic committed 2 minutes ago	
	adding images for intro
 natalijamitic committed 2 minutes ago	
	appending list of emails
 natalijamitic committed 2 minutes ago	
	document changes added XY
 natalijamitic committed 3 minutes ago	
	doc1
 natalijamitic committed 3 minutes ago	
-o-	End of commit history for this file

.gitignore

- some files should not be tracked (e.g., `__pycache__`)
- some files should NEVER be tracked (e.g., API keys, `.env` files)



```
▶  .gitignore
 1 # Demo/example
 2
 3 *.zip
 4 *.gz
 5 log/*.log
 6 log/*.log.[0-9]
 7 assets/videos/
 8 !assets/video/iphone_*.mp4
 9
```

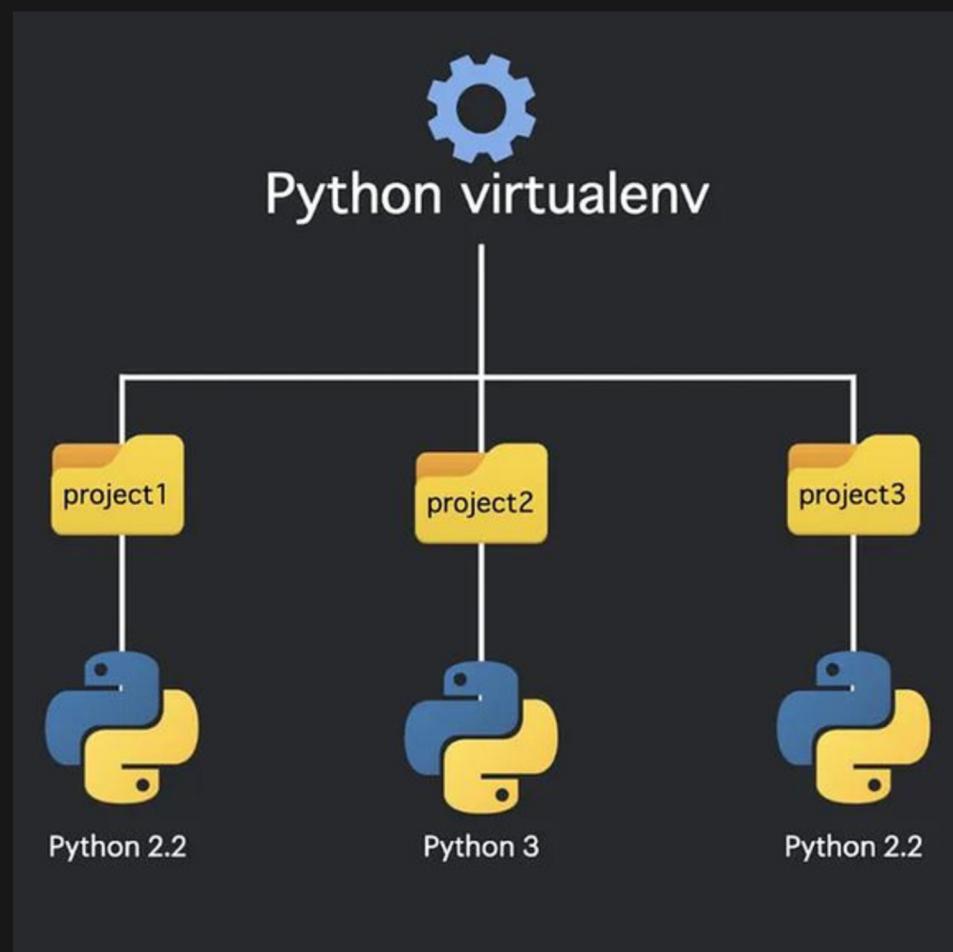


```
⚙ .env      X
Users > gregei > Documents > ⚙ .env
 1 CACHE_ENDPOINT = https://app.org/cache
 2 DATABASE_USERNAME = app_sevg
 3 DATABASE_PASSWORD = asdfmioafjionio2nonf23noiniasndfoin2323
 4 DATABASE_URL = https://db.app.com
```

What is a virtual environment?

- avoid conflicts
- project isolation
- virtualenv:

```
pip install virtualenv  
virtualenv new_env_name  
source new_env_name/bin/activate # (on Mac)
```



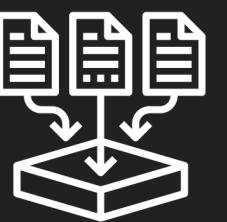
requirements.txt

- pip install -r requirements.txt
- pip freeze | grep name_of_lib

≡ requirements.txt

```
1 # It is possible to specify requirements as plain names.  
2 pytest  
3 pytest-cov  
4  
5 # or with exact version  
6 docopt == 0.6.1  
7  
8 # equivalent to >= 0.23.7, == 0.23.*  
9 pytest-asyncio~=0.23.7  
10  
11 # It is even possible to include other requirement files.  
12 -r other-requirements.txt
```

Building blocks



data loading



model training



model evaluation



testing

Key Takeaways



git and .gitignore
GitHub



python environments
virtualenv