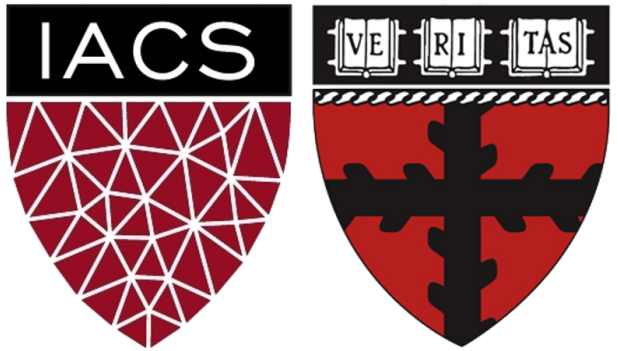
Lecture 1: Introduction: Virtual Environments, Virtual Machines

AC295

Advanced Practical Data Science

Pavlos Protopapas



Outline

1 : Why you should take this class and why not?

2: Who are we?

3: Course structure and activities?

4: Class organization (Workload, Logistics, Grades).

5: Virtual environments.

6: Virtual machines.

Outline

1 : **Why you should take this class and why not?**

2: Who are we?

3: Course structure and activities?

4: Class organization (Workload, Logistics, Grades).

5: Virtual environments.

6: Virtual machines.

Why you should take this class

Because you want to learn how to:

- Put your model in production
- Integrate and orchestrate applications
- Deploy increasing amount of data
- Take advantage of available models
- Evaluate and debug model using visualization

If you have attended **ComputeFest 2020** and found the topics interesting, this class will also be interesting.

Why you shouldn’t take this class

You are **not** familiar with most of the concepts covered in CS109A/B

For example:

- Basic Machine Learning
- CNNs, RNNs, Autoencoders, GANs, etc
- Basic linux commands

Data Science Series to Real World

Data Science Series 109A/B

Ask Question	
Collect Data	CSV file, images, scraping
EDA	Notebook
Methodology	Multiple tasks
Story-telling	Webpage, blogs, posts

Real World

Manage larger database

Learn packages to process larger amount of data

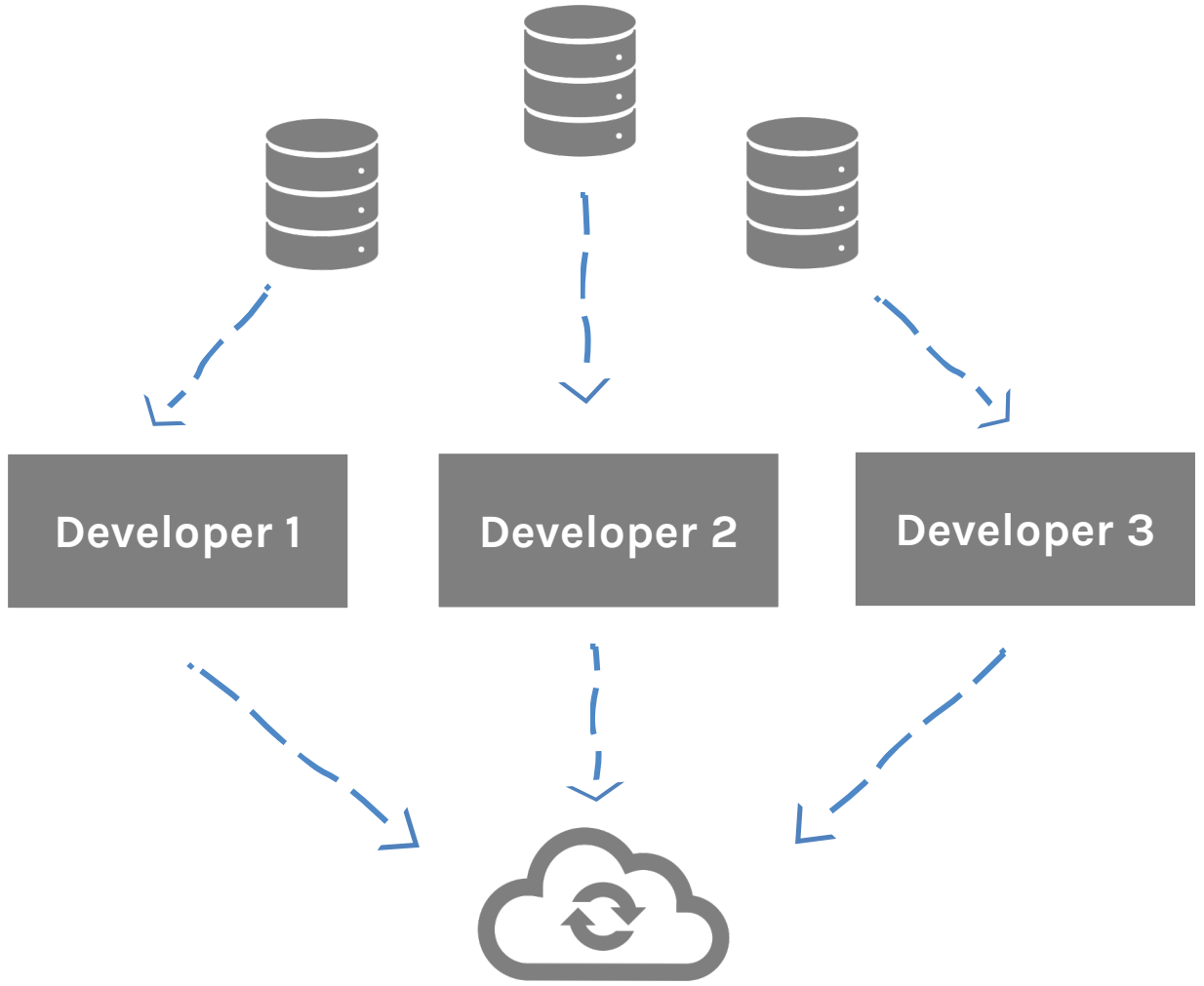
Handle complex team dynamics and orchestrate applications

Data Science Series to Real World (cont)

Fragmented database

Multitude requirements and applications

Recombine and deploy

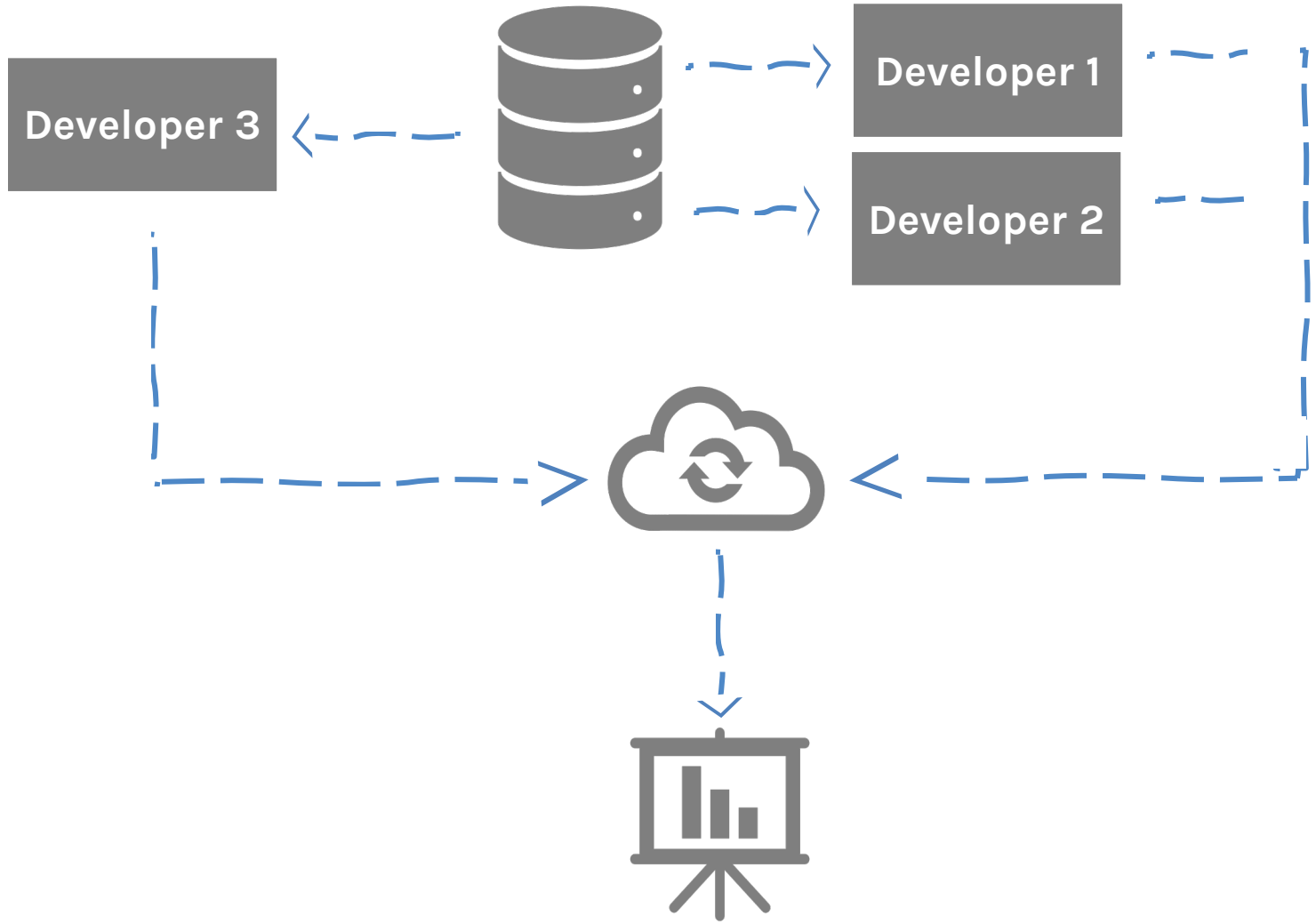


Data Science Series to Real World (cont)

Multiple tasks or models (i.e. Ensemble)

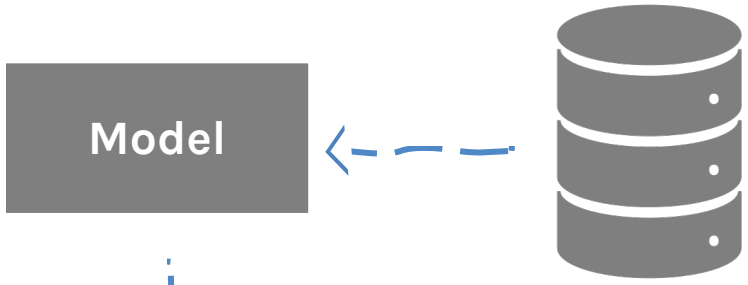
Recombine results

Present results

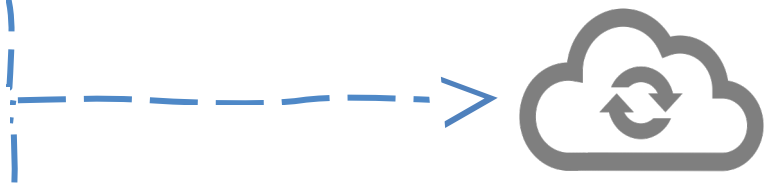


Data Science Series to Real World (cont)

Model too expensive to train
Or not enough training data



Use pre-trained model



Final Results

Present results

Outline

1 : Why you should take this class and why not?

2: Who are we?

3: Course structure and activities?

4: Class organization (Workload, Logistics, Grades).

5: Virtual environments.

6: Virtual machines.

Who?

Pavlos Protopapas

Teaches CS109(a/b), the data science capstone course, and AC295 (advanced practical data science). Research in astrostatistics: machine learning, statistical learning, big data for astronomical problems.

He has picked some new hobbies besides 109s and **eating**:

Going to BSO (well not anymore), cross country ski (completed Engadin skimarathon cheese making and being a TikToker (check me out @pavlosprotopapas)

Who? (cont)

Rashmi Banthia

TF for many Data Science classes here at Harvard including CS109A/B.

Yujiao Chen

TFed for CS109A/B. Currently a Data Scientist

Hai Bui

Graduate Student from Bocconi University in Milan, currently (not) visiting MIT.

Javid Lakha

Machine Learning Engineer at Legatics (a legal technology start-up).

Who? (cont)

Shivas Jayaram

CTO and Co-Founder @ Brain Cradle.

Andrea Porelli

Master's from IACS CSE.

William Palmer

Data Science student at IACS.

Faras Sadek

Outline

1 : Why you should take this class and why not?

2: Who are we?

3: **Course structure and activities?**

4: Class organization (Workload, Logistics, Grades).

5: Virtual environments.

6: Virtual machines.

Course Structure and Activities

Modules:

1. Deploy data science (integration + scalability)
2. Transfer learning and distillation
3. Visualization as investigative tool * [no presentations or exercises]

Activities:

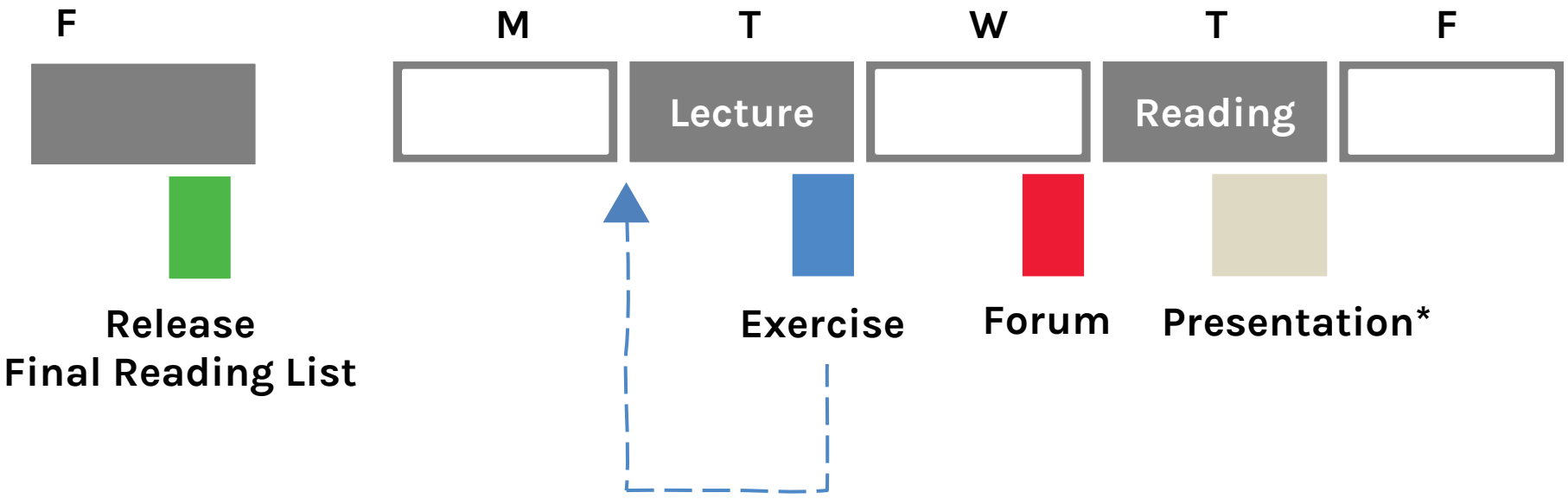
lectures, reading and presentations, exercises, forum, practicums, projects

Lectures online: Tuesdays 10:30-11:45 am (repeat 6:00-7:15 pm)

Presentations on Reading and Discussions: Thursdays 10:30-11:45 am (repeat 6:00-7:15 pm)

Course Structure and Activities

Regular week schedule



Topics

Deploy data science (integration + scalability)

A. Virtual Environments, Virtual Boxes, and Containers

B. Kubernetes

C. Dask

Topics (cont)

Transfer learning and distillation

A. Intro to Transfer Learning: basics and Convolutional Neural Networks review

B. Transfer Learning across Tasks for images and SOTA Models

C. Language Models and Transfer Learning with Text Data

D. Attention and Transformers

E. Distillation and Compression

Topics (cont)

Visualization as investigative tool

A. Introduction and Overview of Viz for Deep Models: lime and shapley

B. CNN for Image Data, Activation Maximization and Saliency Maps

C. Attention for Debugging Language Models

Calendar

> *Link to Calendar* <

Week	Date	Lecture #	Topics	Exercise
1	9/3	1	Introduction: Virtual Enviroments and Virtual Boxes	
2	9/8	2	Containers	EX1
	9/10		Use Case: Dockers in a real setting	
3	9/15	3	Kubernetes	EX2
	9/17		Use Case: Kubernetes in a real setting	
4	9/22	4	Dask	EX3
	9/24		Use Case: Dask in a real setting	
5	9/29		Practicum 1: End to end art search engine	Practicum 1
	10/1		Practicum 1	
6	10/6	5	Intro to Transfer Learning: basics and CNNs review	EX4
	10/8		Journal Discussion: Transfer Learning (Statistical approaches to Transfer Learning)	
7	10/13	6	Transfer Learning for Images and SOTA Models	EX5
	10/15		Journal Discussion:	
8	10/20	7	Language Models and Transfer Learning for Text	EX6
	10/22		Journal Discussion	
9	10/27	8	Attention and Transformers	EX7
	10/29		Journal Discussion	
10	11/3	9	Distillation and Compression	EX8
	11/5		Journal Discussion	
11	11/10		Practicum 2	Practicum 2
	11/12		Practicum 2	
12	11/17	10	Introduction and Overview of Viz for Deep Models: lime and shapley	
	11/19	11	CNNs for Image Data, Activation Maximization and Saliency Maps	
13	11/24	12	Attention for Debugging Language Models	
	11/26			
14	12/1		Project	
	12/3		Project	
15	12/8		Project	
	12/11		Final projects presentation	

Outline

1 : Why you should take this class and why not?

2: Who are we?

3: Course structure and activities?

4: **Class organization (Workload, Logistics, Grades).**

5: Virtual environments.

6: Virtual machines.

Workload

Regular Week

3 hours in class
5 hours reading
5 hours exercise
1 hour forum questions
3 hours presentation*

~ 16 hours/week

* 1 presentation per module per group (2 total)

Practicum and Project Week

~ 16 hours/week**

** 2 practicums and 1 final project (2 weeks long)

We will be asking for your feedback on the workload

Expectations

How to read and present class material

> *Link to Reading Guidelines* <

> *Link to Presentation Guidelines* <

Logistics

Fill up forms

Survey

Make group *

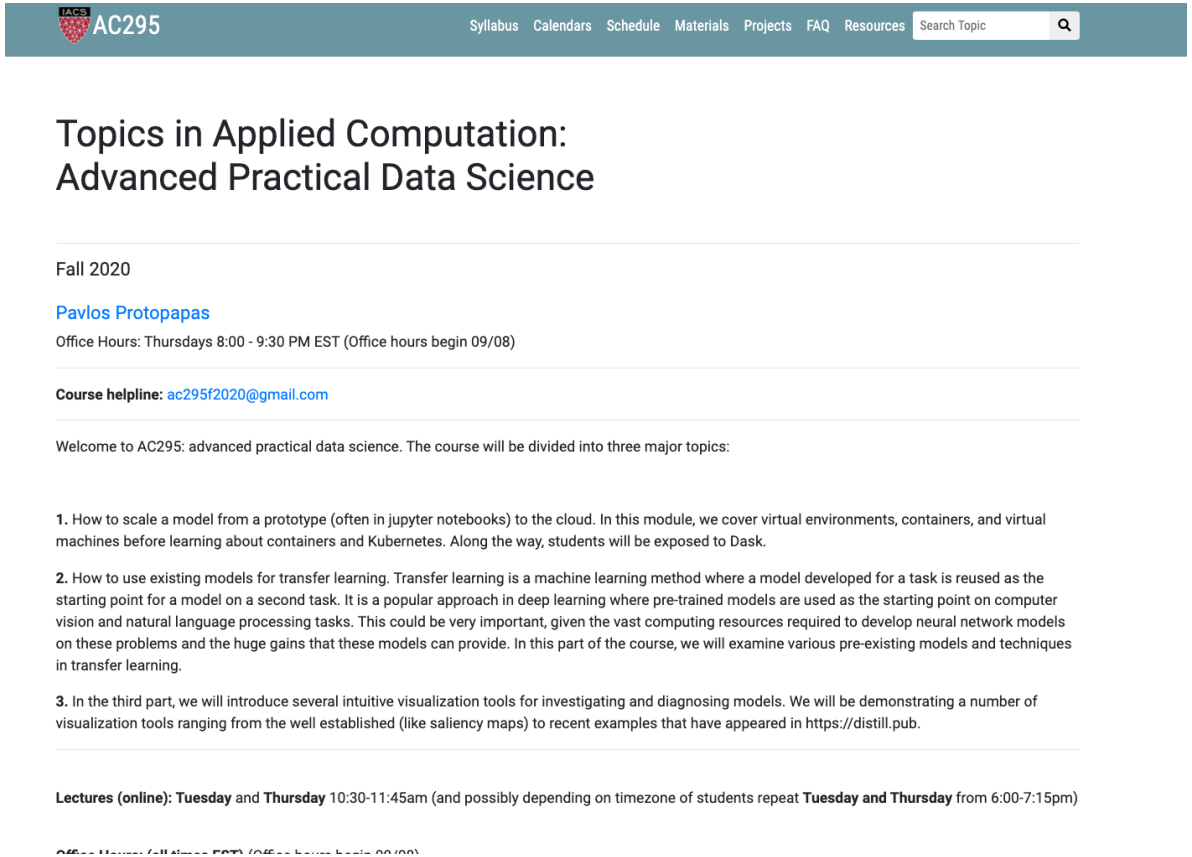
Sign-up presentation**

* Fill group components in each row

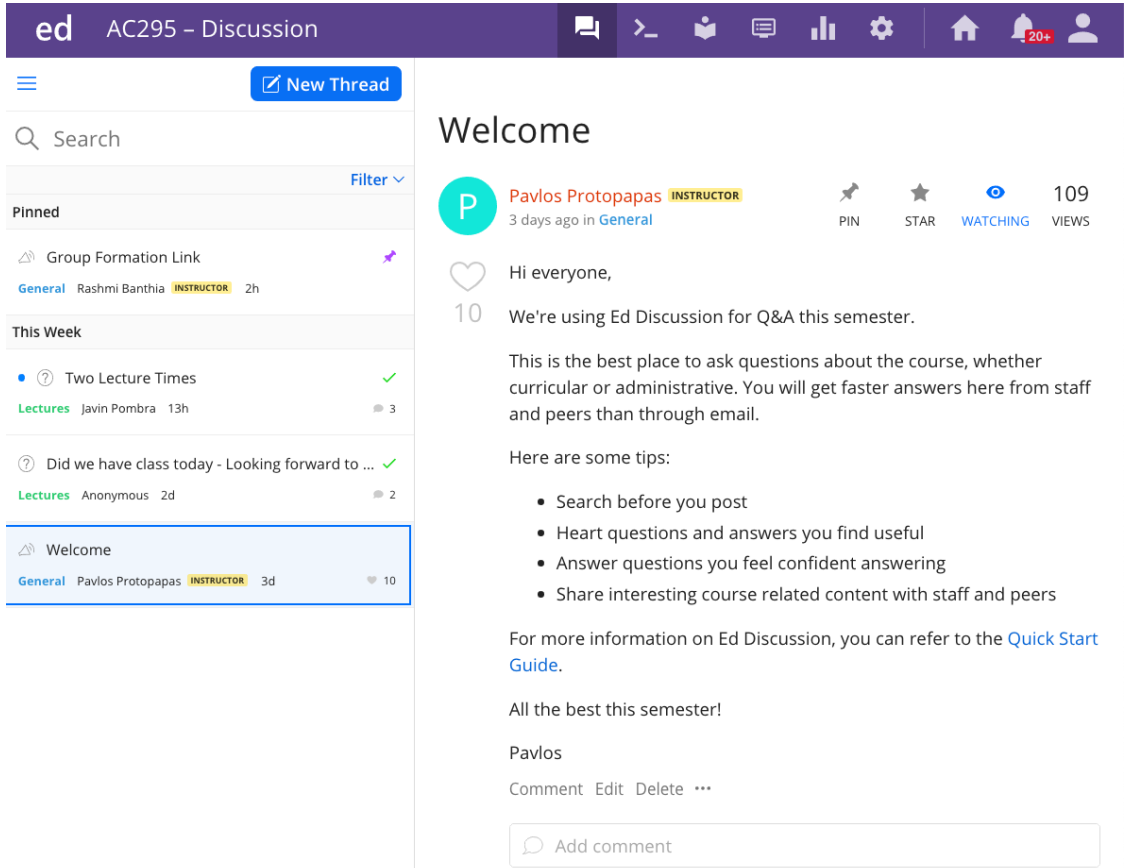
** Each group should pick one slot (white background) in each module. We will release presentation slots on Sunday 8PM

Course Components

Web Page: Syllabus, lecture slides and notebooks



Edstem: Forum and surveys



Github Repo: https://github.com/Harvard-IACS/2020F-AC295.git

Grades

Assignment	Final Grade Weight
Discussion Forum	10%
Exercises	10%
Presentations	15%
Practicums	40%
Final Projects	25%
Total	**100%**

Final Details

- We will be using ED for discussions, announcements and surveys.

- Individual,

  Submit at Canvas

- Presentations: Group
- Practicums: Group
- Projects: Group

  Submissions for presentations, practicums and projects we will be using github (details soon).

Outline

1 : Why you should take this class and why not?

2: Who are we?

3: Course structure and activities?

4: Class organization (Workload, Logistics, Grades).

5: Virtual environments.

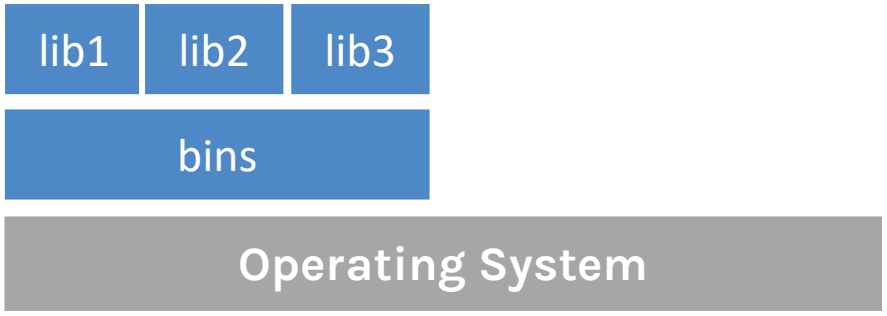
6: Virtual machines.

Why should we use virtual environment?

- Virtual environments help to make development and use of code more streamlined.
- Virtual environments keep dependencies in separate “sandboxes” so you can switch between both applications easily and get them running.
- Given an operating system and hardware, we can get the exact code environment set up using different technologies. This is key to understand the trade off among the different technologies presented in this class.

Why should we use virtual environment?

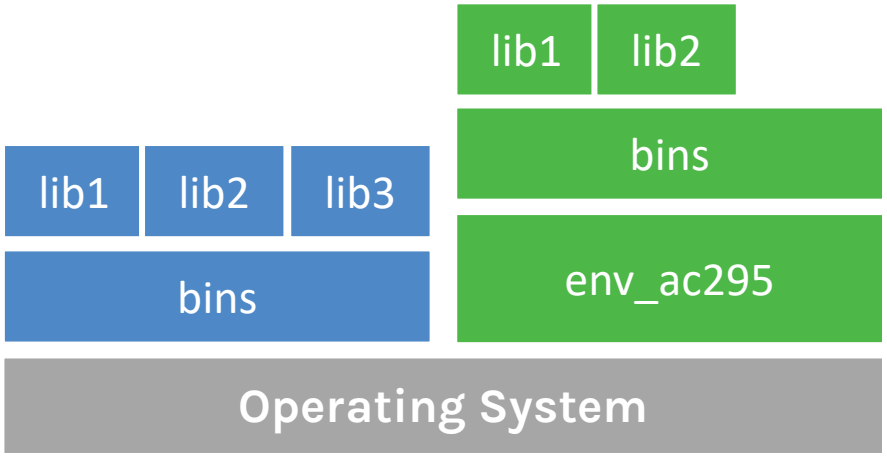
- Maggie took cs109a, she used to run her Jupyter notebooks from anaconda prompt. Every time she installed a module it was placed in the either of `bin, lib, share, include` folders and she could import it in and used it without any issue.




```
$ which python
/c/Users/maggie/Anaconda3/python
```

Why should we use virtual environment?

- Maggie starts taking ac295, and she thinks that it would be good to isolate the new environment from the previous environments avoiding any conflict with the installed packages. She adds a layer of abstraction called virtual environment that helps her keep the modules organized and avoid misbehaviors while developing a new project.

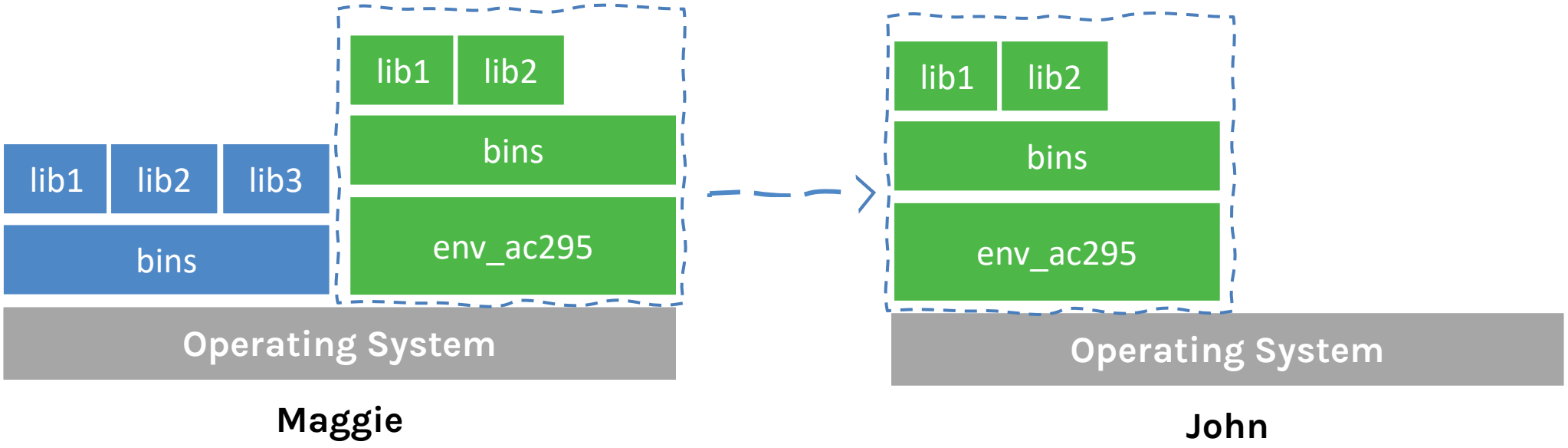


Maggie

```
$ which python
/c/Users/maggie/Anaconda3/envs/env_ac295/python
```

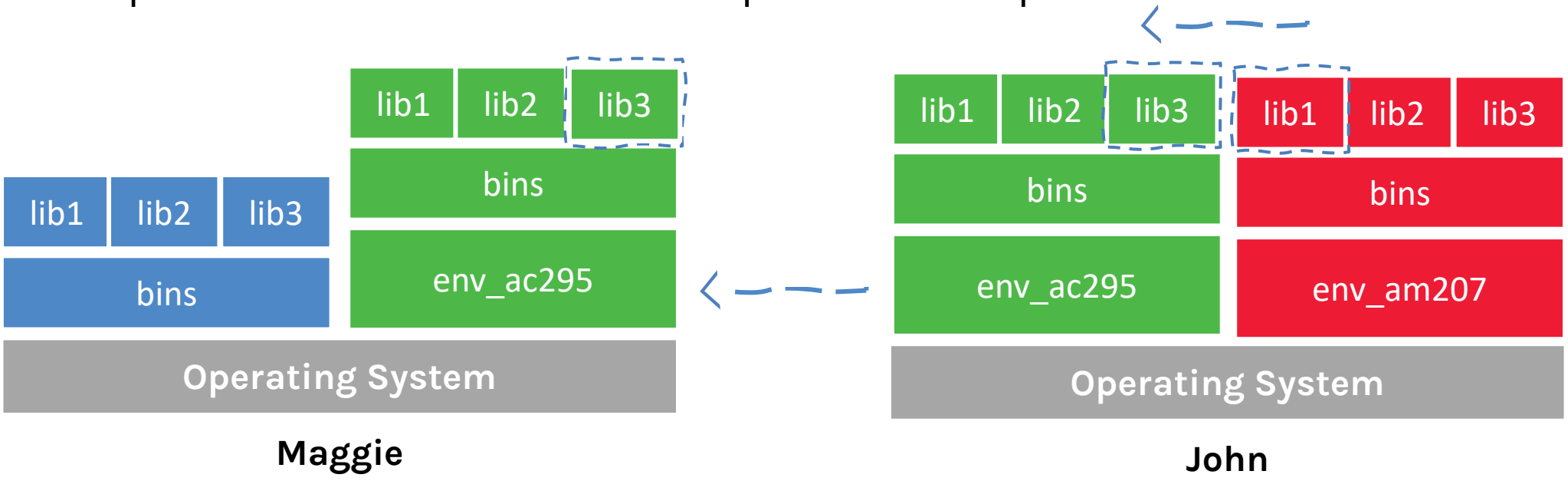
Why should we use virtual environment?

- Maggie collaborates with John for the final project and shares the environment she is working on through .yml file.



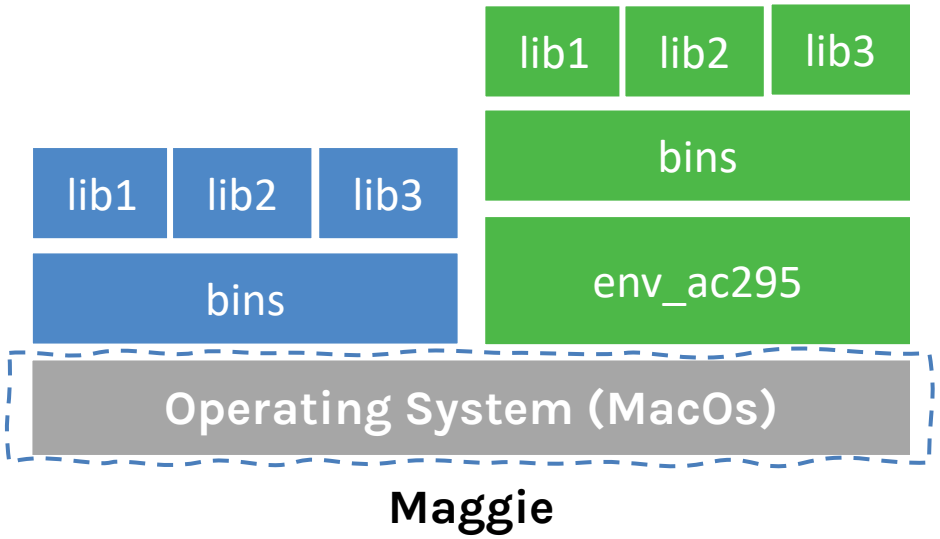
Why should we use virtual environment?

- John experiments a new method he learned in another class and adds a new library to the working environment. After seeing tremendous improvements, he sends Maggie back his code and a new .yml file. She can now update her environment and replicate the experiment.

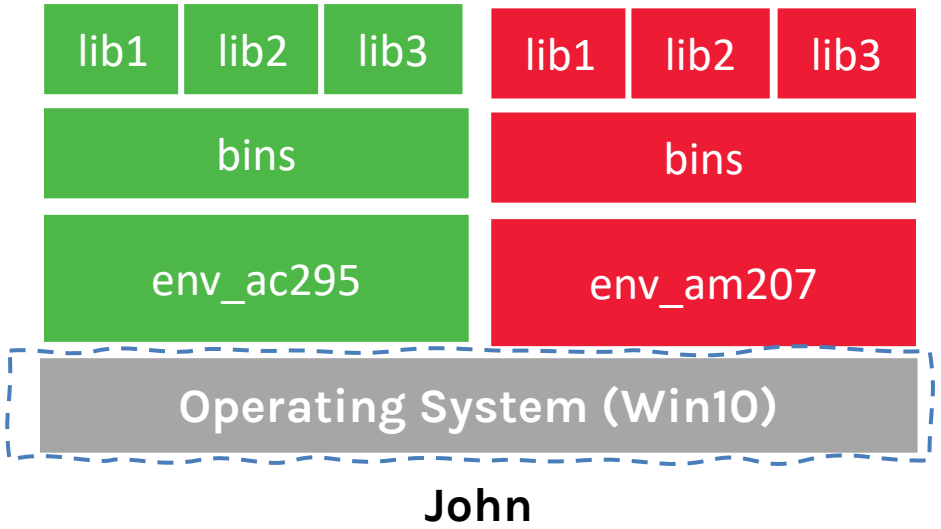


Why should we use virtual environment?

- What could go wrong? Unfortunately, Maggie and John reproduce different results, and they think the issue relates to their operating systems. Indeed while Maggie has a MacOS, John uses a Win10.



Maggie



John

Virtual environments

Pros

- Reproducible research
- Explicit dependencies
- Improved engineering collaboration
- Broader skill set

Cons

- Difficulty setting up your environment
- Not isolation
- Does not work across different OS

What are virtual environments then?

A virtual environment is a directory with the following components:

- site_packages/ directory where third-party libraries are installed
- links [really symlinks] to the executables on your system
- some scripts that ensure that the code uses the interpreter and site packages in the virtual environment

> *Adapted from CS207* <

Virtual environments: virtualenv vs conda

virtualenv

- virtual environments manager embedded in Python
- incorporated into broader tools such as `pipenv`
- allow to install modules using pip package manager

how to use **virtualenv**

- create an environment within your project folder `virtualenv your_env_name`
- it will add a folder called environment_name in your project directory
- activate environment: `source env/bin/activate`
- install requirements using: `pip install package_name=version`
- deactivate environment once done: `deactivate`

Virtual environments in practice

conda environment

- virtual environments manager embedded in Anaconda
- allow to use both conda and pip to manage and install packages

how to use **conda**

- create an environment

```
conda create --name your_env_name python=3.7
```

- it will add a folder located within your anaconda installation

```
/Users/your_username /anaconda3/envs/your_env_name
```

- activate environment `conda activate your_env_name` (should appear in your shell)
- install requirements using `conda install package_name=version`
- deactivate environment once done `conda deactivate`
- duplicate your environment using YAML file `conda env export > my_environment.yml`
- to recreate the environment now use `conda env create -f environment.yml`

how to use **conda**

- find which environment you are using

  `conda env list`

- create an environment

  `conda create --name your_env_name python=3.7`

- it will add a folder located within your anaconda installation

  `/Users/your_username/[opt]/anaconda3/envs/your_env_name`

- activate environment

  `conda activate your_env_name` (should appear in your shell)

- install requirements using

  `conda install package_name=version`

- deactivate environment once done

  `conda deactivate`

- duplicate your environment using YAML file `conda env export > my_environment.yml`
- to recreate the environment now use `conda env create -f environment.yml`

More on Virtual environments

Further readings

- For detailed discussions on similarities and differences among virtualenv and conda https://jakevdp.github.io/blog/2016/08/25/conda-myths-and-misconceptions/
- More on venv and conda environments
  https://towardsdatascience.com/virtual-environments-104c62d48c54
  https://towardsdatascience.com/getting-started-with-python-environments-using-conda-32e9f2779307

Outline

1 : Why you should take this class and why not?

2: Who are we?

3: Course structure and activities?

4: Class organization (Workload, Logistics, Grades).

5: Virtual environments.

6: Virtual machines.

Why should we use virtual machines?

Motivation

- We have our isolated systems, and after we set up the environment with our colleagues' machine, we expect to get identical results, right? Unfortunately, it is not always the case. Why? Most likely because we run on a different operating system.
- Even though using virtual environments, we isolate our computations, we might need to use the same operating system that requires running "like if" we are in different machines.
- How can we run the same experiment? Virtual Machines!
- Isolation!

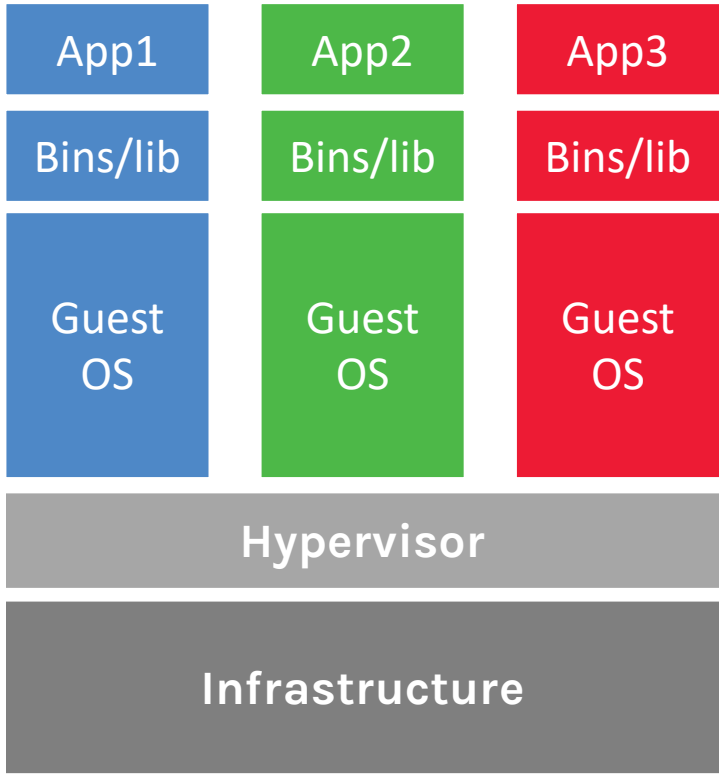
Why should we use virtual machines? (cont)

Advantages

- Full autonomy: it works like a separate computer system; it is like running a computer within a computer.
- V**ery secure**: the software inside the virtual machine cannot affect the actual computer.
- Lower costs: buy one machine and run multiple operating systems.

What are virtual machines?

- virtual machines have their own virtual hardware: CPUs, memory, hard drives, etc.
- you need a hypervisor that manages different virtual machines on server
- hypervisor can run as many virtual machines as you wish
- operating system is called the "host" while those running in a virtual machine are called "guest"
- You can install a completely different operating system on this virtual machine



Machine Virtualization

https://towardsdatascience.com/how-to-install-a-free-windows-virtual-machine-on-your-mac-bf7cbc05888e

Limitations

- Uses hardware in your local machine
- There is an overhead associated with virtual machines
    1. *Guest* is not as fast as the *host* system
    2. Takes a long time to start up
    3. It may not have the same graphics capabilities

This is the second time we are offering the course, so your feedback will improve it for future years.

However, we are making every effort to have a well-organized course and we promise you an exciting semester full of learning!

THANK YOU