

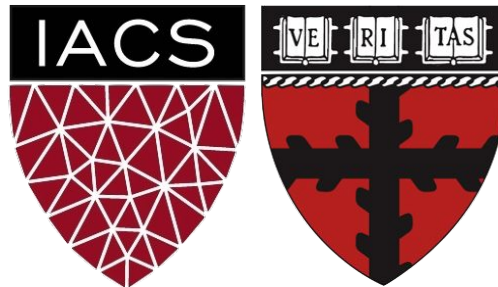
Lecture 12: Virtual Environments & Virtual Machines

Advanced Practical Data Science, MLOps

AC295

Pavlos Protopapas

Institute for Applied Computational Science, Harvard



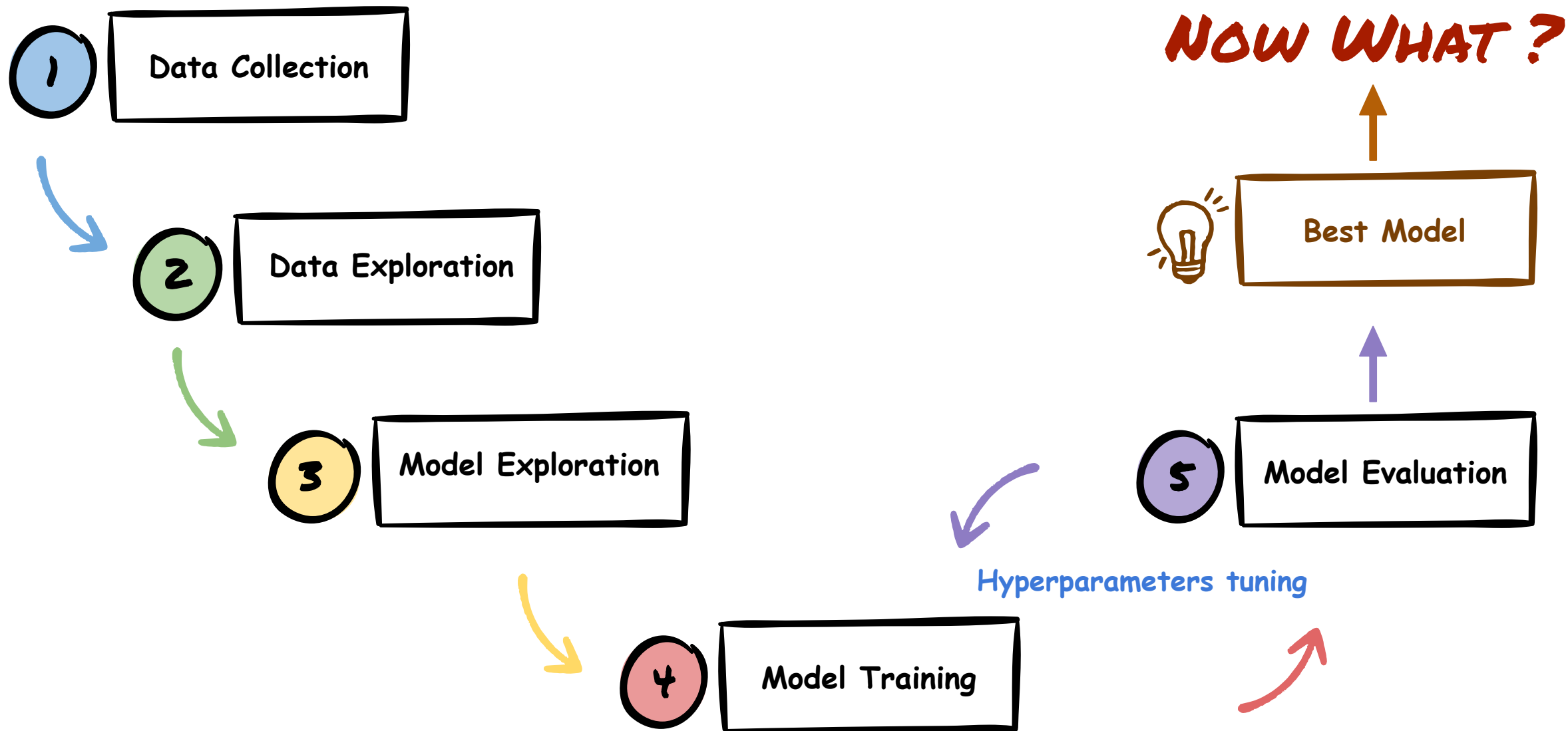
Outline

1. Where we are
2. Motivation
3. Virtual Machines
4. Virtual Environments

Outline

1. **Where we are**
2. Motivation
3. Virtual Machines
4. Virtual Environments

Where we are: Deep Learning Flow



Leaderboard - Best Model

 **Mushroom Classification Leaderboard** 

Total Models Submitted: 591

	User	Score *	Model Size (Mb) **	Accuracy ***	Total Parameters	Accuracy (Reported)	Loss (Reported)
1	archclojure@gmail.com	0.8844	8.73	84.33%	2,340,358	80.24%	2.25
2	neil_sehgal@g.harvard.edu	0.8799	22.33	85.00%	6,001,686	79.29%	1.09
3	mbavare21@gmail.com	0.8774	22.64	84.67%	6,084,054	86.05%	2.50
4	kevinhare@g.harvard.edu	0.875	22.63	84.33%	6,084,054	81.87%	1.49
5	anita.mahinpei@gmail.com	0.8681	8.72	82.00%	2,340,358	75.37%	1.44
6	kxyang@ucdavis.edu	0.8659	22.33	83.00%	6,001,686	77.56%	0.49
7	wangyuanbiao2016@gmail.com	0.8563	26.19	82.00%	7,047,750	91.18%	0.69
8	bharpe@college.harvard.edu	0.8494	8.73	79.33%	2,340,358	75%	2.43

Outline

1. Where we are
- 2. Motivation**
3. Virtual Machines
4. Virtual Environments

We want to build a 🍄 Mushroom Finder App

- Pavlos likes to go the forest for mushroom picking
- Some mushrooms can be poisonous
- Help build an app to identify mushroom type and if poisonous or not

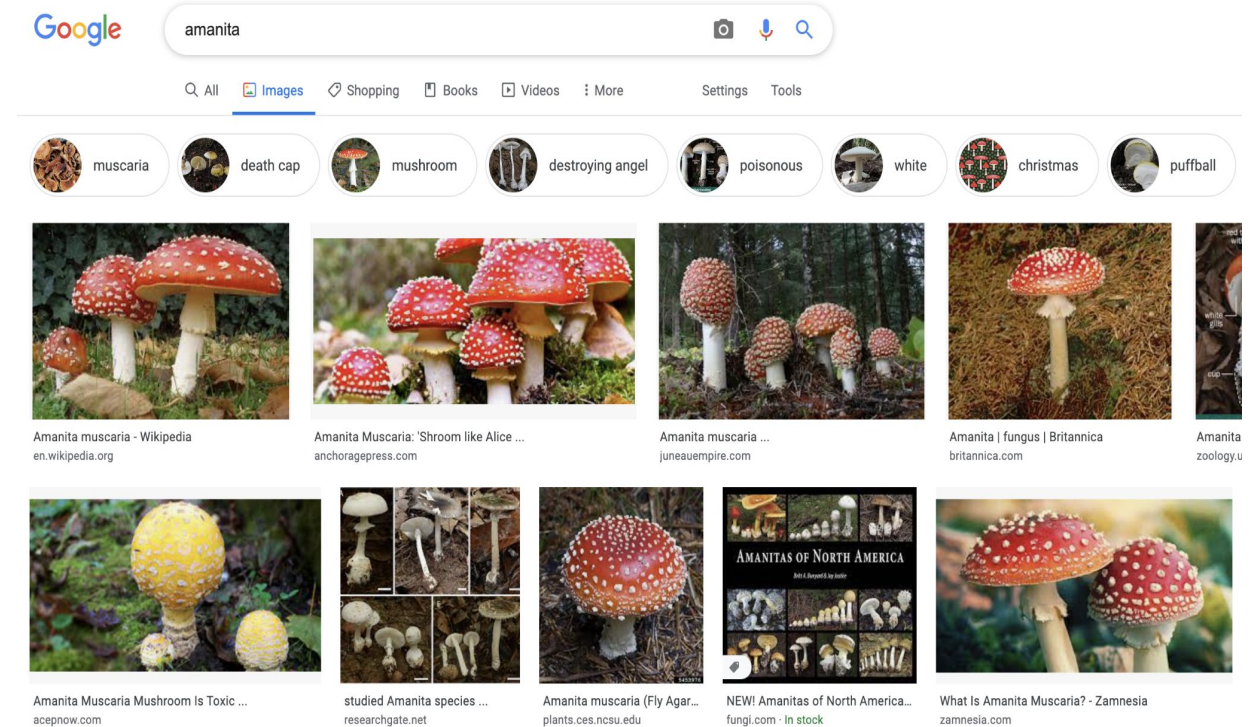


Credit: Nikolas Protopapas



Mushroom App: Data Collection

- Collect images from Google
- For our demo we downloaded images for mushrooms **oyster**, **crimini**, **amanita (Poisonous)**



Python Script



Mushroom App: Models

- Identify our problem task
- Try various model architectures
- Transfer Learning
- Hyperparameters tuning
- In class competition

Mushroom Classification Leaderboard 🍄

Total Models Submitted: 591

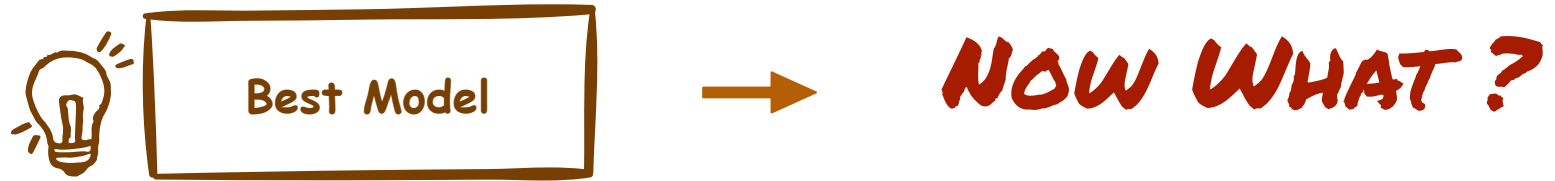
	User	Score *	Model Size (Mb) **	Accuracy ***	Total Parameters	Accuracy (Reported)	Loss (Reported)
1	archclojure@gmail.com	0.8844	8.73	84.33%	2,340,358	80.24%	2.25
2	neil_sehgal@g.harvard.edu	0.8799	22.33	85.00%	6,001,686	79.29%	1.09
3	mbavare21@gmail.com	0.8774	22.64	84.67%	6,084,054	86.05%	2.50
4	kevinhare@g.harvard.edu	0.875	22.63	84.33%	6,084,054	81.87%	1.49
5	anita.mahinpei@gmail.com	0.8681	8.72	82.00%	2,340,358	75.37%	1.44
6	kxyang@ucdavis.edu	0.8659	22.33	83.00%	6,001,686	77.56%	0.49
7	wangyuanbiao2016@gmail.com	0.8563	26.19	82.00%	7,047,750	91.18%	0.69
8	bharpe@college.harvard.edu	0.8494	8.73	79.33%	2,340,358	75%	2.43

	trainable_parameters	execution_time	loss	accuracy
2	2,388,227	3.24 mins	82.65	88.48%
3	2,306,051	3.24 mins	42.84	87.27%
1	164,355	2.31 mins	82.45	86.67%
4	6,001,686	3.24 mins	42.84	79.39%
5	2,340,358	3.24 mins	42.84	64.24%
6	6,001,686	3.24 mins	42.84	63.03%
7	7,047,750	3.24 mins	42.84	63.03%

Colab



Mushroom App: Best Model





Mushroom App

- We want to build an app to take a photo of a mushroom and it helps us identify the type of mushroom
- How do we build the app?



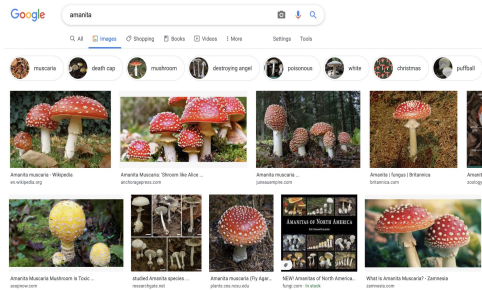
Type: amanita (93.54%)

How do we build an App?

- Collaborate with team to **design** and **develop**
- Expose best model as an **API**
- Build a **frontend** using HTML & javascript
- **Deploy** app to a cloud provider
- <http://awesome-mushroom-app.com> [Go live]

How do we build an App?

Data Collection



Python Script



Data Exploration
Model Exploration
Model Training
Model Evaluation

04_demo_mushroom_classification_models.ipynb

	trainable_parameters	execution_time	loss	accuracy	model_size
0	164,355	4.26 mins	70.76	89.70%	10 MB
5	2,306,051	2.71 mins	42.90	87.88%	10 MB
2	2,388,227	2.78 mins	82.48	87.27%	10 MB
4	82,179	2.56 mins	42.84	86.67%	10 MB
6	25,950,531	7.44 mins	0.91	66.06%	104 MB
3	11,112,323	7.58 mins	0.79	61.21%	45 MB
1	22,514,755	4.70 mins	1.03	46.06%	90 MB

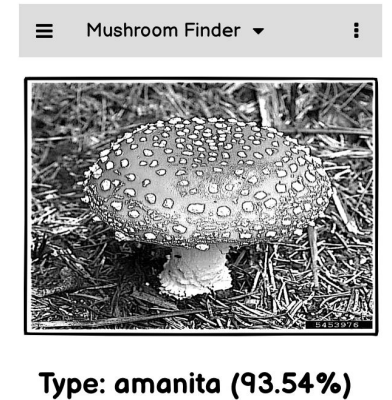
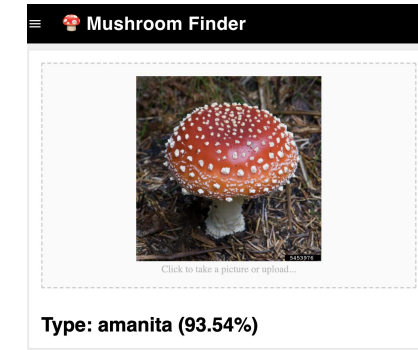
Colab



Rest API

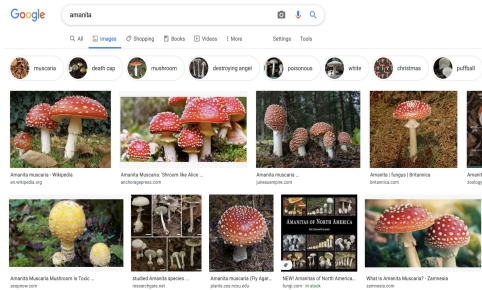
Best Model

IDE / Code Editor



How do we build an App?

Data Collection



Python Script

Data Exploration
Model Exploration
Model Training
Model Evaluation

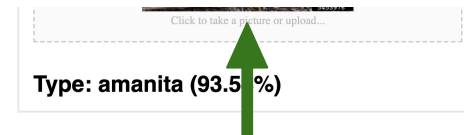
04_demo_mushroom_classification_models.ipynb

	trainable_parameters	execution_time	loss	accuracy	model_size
0	164,355	4.26 mins	70.76	89.70%	10 MB
5	2,306,051	2.71 mins	42.90	87.88%	10 MB
2	2,388,227	2.78 mins	82.48	87.27%	10 MB
4	82,179	2.56 mins	42.84	86.67%	10 MB
6	25,950,531	7.44 mins	0.91	66.06%	104 MB
3	11,112,323	7.58 mins	0.79	61.21%	45 MB
1	22,514,755	4.70 mins	1.03	46.06%	90 MB

Colab

Mushroom Finder

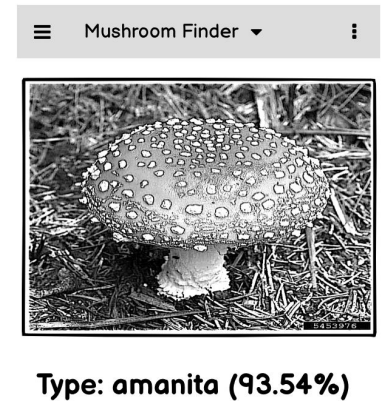
PRODUCTIONIZING MODEL !



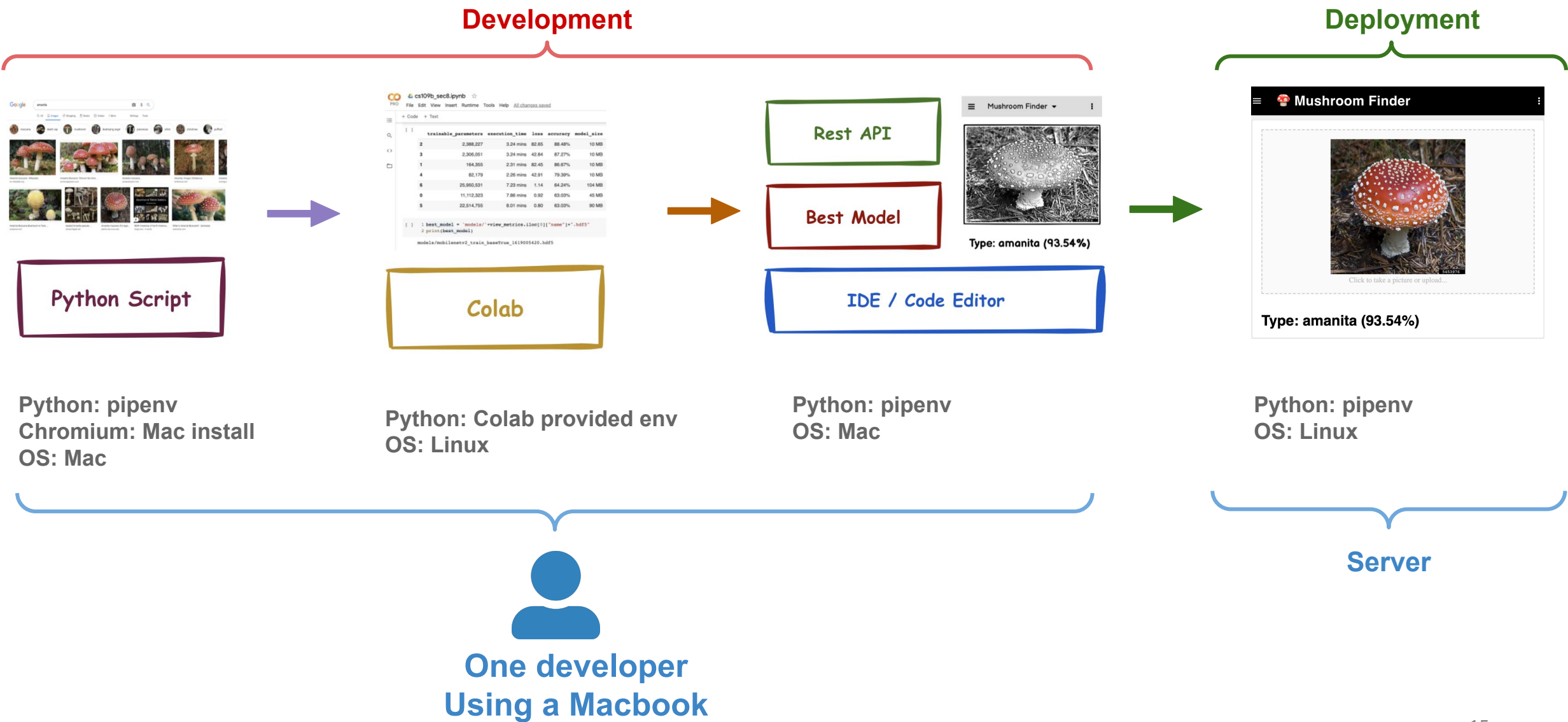
Rest API

Best Model

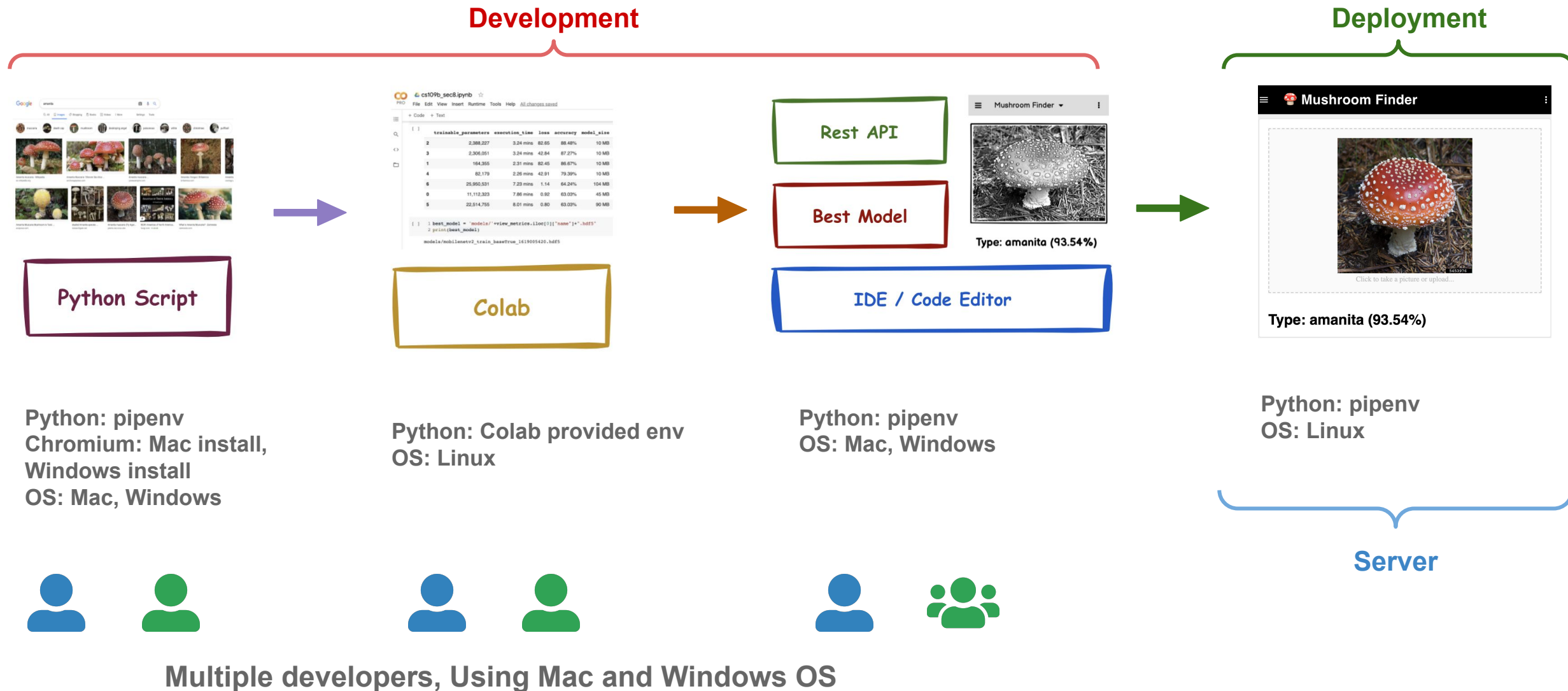
IDE / Code Editor



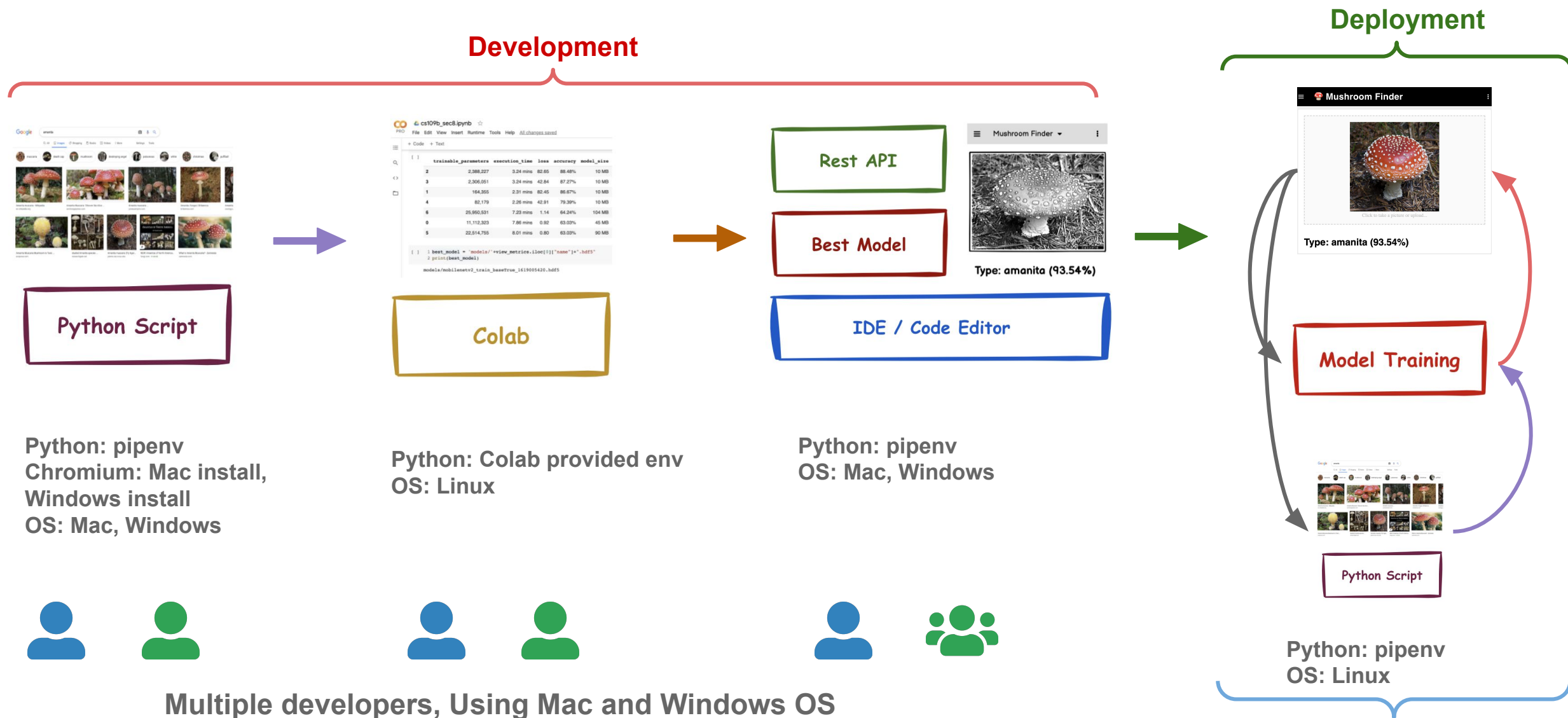
Challenges



Challenges - Multiple Developers



Challenges - Multiple Developers + Automation



Challenges / Solutions

Challenges:

- OS specific installations required
- How to collaborate code/frameworks?
- Automate data collection / model training
- New team member onboarding
- “It works on my machine” ͇_(ツ)_/͇

Solutions:

- Isolate development into environments that can be shared
- Develop in an common OS regardless of developers host OS
- Track software/framework installs

Tools

- Virtual Machines
- Virtual Environments
- Containers

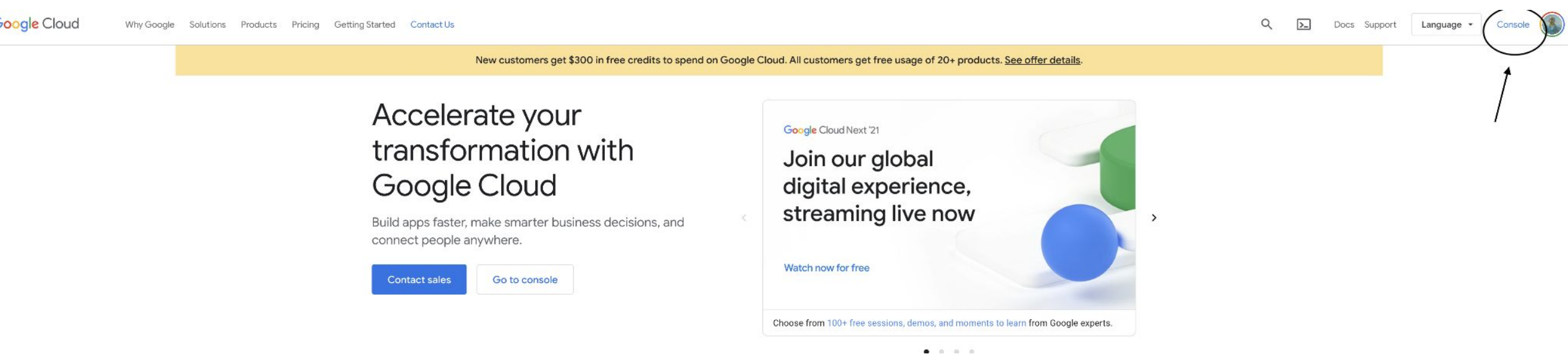
Outline

1. Where we are
2. Motivation
- 3. Virtual Machines**
4. Virtual Environments

Virtual Machines Demo

<https://cloud.google.com/billing/docs/how-to/edu-grants>

<https://cloud.google.com/>



The screenshot shows the Google Cloud homepage. At the top left is the Google Cloud logo. To its right is a navigation menu with links: Why Google, Solutions, Products, Pricing, Getting Started, and Contact Us. On the far right of the top navigation bar are search, a hamburger menu icon, Docs, Support, a Language dropdown, and a 'Console' link. The 'Console' link is circled in black with a hand-drawn arrow pointing to it from the right. Below the navigation bar is a yellow banner with the text: 'New customers get \$300 in free credits to spend on Google Cloud. All customers get free usage of 20+ products. [See offer details.](#)' The main content area features a large heading 'Accelerate your transformation with Google Cloud' on the left, followed by the subtext 'Build apps faster, make smarter business decisions, and connect people anywhere.' Below this are two buttons: 'Contact sales' and 'Go to console'. To the right of this text is a carousel slide for 'Google Cloud Next '21' with the headline 'Join our global digital experience, streaming live now' and a 'Watch now for free' link. The slide includes a background image of a blue sphere and a green cylinder. At the bottom of the slide, it says 'Choose from 100+ free sessions, demos, and moments to learn from Google experts.'

Virtual Machines Demo

Go to Navigation Menu

Google Cloud Platform

Preparing For Class

Navigation menu

ACTIVITY

RECOMMENDATIONS

Search products and resources

CUSTOMIZE

Google Cloud Next '21 is live. Join us now at g.co/cloudnext. DISMISS

Project info

Project name
Preparing For Class

Project ID
preparing-for-class

Project number
406706675980

[ADD PEOPLE TO THIS PROJECT](#)

[Go to project settings](#)

API APIs

Requests (requests/sec)

1.0
0.8
0.6
0.4
0.2
0

No data is available for the selected time frame.

Google Cloud Platform status

All services normal

[Go to Cloud status dashboard](#)

Billing

Estimated charges
For the billing period Oct 1 - 12, 2021

USD \$0.00

Virtual Machines Demo

Select compute engine

The screenshot displays the Google Cloud Platform (GCP) console interface. On the left, a navigation sidebar is visible with the following sections:

- Home**: Includes links to Home, Recent, Cloud Storage, Compute Engine, and Kubernetes Engine.
- PRODUCTS**: Includes Marketplace, Billing, APIs & Services, Support, IAM & Admin, Getting started, Compliance, Security, and Anthos.
- COMPUTE**: Includes Compute Engine (highlighted with a blue pin icon).

An arrow points from the 'Compute Engine' option in the 'HOME' section to the main content area. The main content area shows the 'Project info' section for the project 'Preparing For Class' (Project ID: preparing-for-class, Project number: 406706675980). Below this, there are sections for 'Resources', 'Trace', and 'Getting Started'. The 'Resources' section indicates that the project has no resources. The 'Trace' section shows no trace data from the past 7 days. The 'Getting Started' section is partially visible at the bottom.

On the right side of the console, there are several informational panels:

- Google Cloud Platform status**: All services normal. Includes a link to 'Go to Cloud status dashboard'.
- Billing**: Estimated charges for the billing period Oct 1 - 12, 2021 are USD \$0.00. Includes a link to 'View detailed charges'.
- Monitoring**: Includes links to 'Create my dashboard', 'Set up alerting policies', 'Create uptime checks', and 'View all dashboards'.

The top of the console features a blue header with the Google Cloud Platform logo, the project name 'Preparing For Class', a search bar, and a 'CUSTOMIZE' button.

Virtual Machines Demo

Select Virtual Machines

Google Cloud Platform

Preparing For Class

Search products and resources

Compute Engine

Virtual machines

VM instances

Instance templates

Sole-tenant nodes

Machine images

TPUs

Committed use discounts

Migrate for Compute Engi...

Storage

Disks

Snapshots

Images

Instance groups

Instance groups

Health checks

VM Manager

OS patch management

OS configuration manage...

Settings

Metadata

VM instances

CREATE INSTANCE

IMPORT VM

REFRESH

START / RESUME

STOP

SUSPEND

RESET

DELETE

CREATE SCHEDULE

Filter


Enter property name or value

?

⋮

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
--------	------	------	-----------------	-----------	-------------	-------------	---------

VM Instances

A graphic for VM Instances featuring a central circle with vertical wavy lines, surrounded by several smaller colored circles (green, red, blue, yellow) and dots.

VM Instances

Compute Engine lets you use virtual machines that run on Google's infrastructure. Create micro-VMs or larger instances running Debian, Windows, or other standard images. Create your first VM instance, import it using a migration service, or try the quickstart to build a sample app.

CREATE INSTANCE

TAKE THE QUICKSTART

Virtual Machines Demo

Select all defaults

Google Cloud Platform

Preparing For Class

Search products and resources

Create an instance

To create a VM instance, select one of the options:

New VM instance

Create a single VM instance from scratch

New VM instance from template

Create a single VM instance from an existing template

New VM instance from machine image

Create a single VM instance from an existing machine image

Marketplace

Deploy a ready-to-go solution onto a VM instance

GENERAL-PURPOSE

COMPUTE-OPTIMIZED

MEMORY-OPTIMIZED

GPU

Machine types for common workloads, optimized for cost and flexibility

Series

E2

CPU platform selection based on availability

Machine type

e2-medium (2 vCPU, 4 GB memory)

vCPU

1 shared core

Memory

4 GB

CPU PLATFORM AND GPU

Display device

Enable to use screen capturing and recording tools.

Enable display device

Confidential VM service

Enable the Confidential Computing service on this VM instance.

Container

Deploy a container image to this VM instance

DEPLOY CONTAINER

Boot disk

Disk type

New balanced persistent disk

Disk size

10 GB

Image

Debian GNU/Linux 10 (buster)

CHANGE

Identity and API access

Service accounts

Service account

Compute Engine default service account

Access scopes

Allow default access

Allow full access to all Cloud APIs

Set access for each API

Monthly estimate

\$25.46

That's about \$0.03 hourly

Pay for what you use: No upfront costs and per second billing

DETAILS

25

Virtual Machines Demo

Wait for instance to start and click on ssh

Google Cloud Platform

Preparing For Class

Search products and resources

Compute Engine

VM instances

CREATE INSTANCE IMPORT VM REFRESH START / RESUME STOP SUSPEND RESET DELETE CREATE SCHEDULE

Virtual machines

VM instances

Instance templates

Sole-tenant nodes

Machine images

TPUs

Committed use discounts

Migrate for Compute Engi...

Storage

Disks

Snapshots

Images

Instance groups

Instance groups

Health checks

VM Manager

INSTANCES INSTANCE SCHEDULE

VM instances are highly configurable virtual machines for running workloads on Google infrastructure. [Learn more](#)

Filter Enter property name or value

<input type="checkbox"/>	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>	✓	instance-1	us-central1-a			10.128.0.7 (nic0)	34.132.242.220	SSH

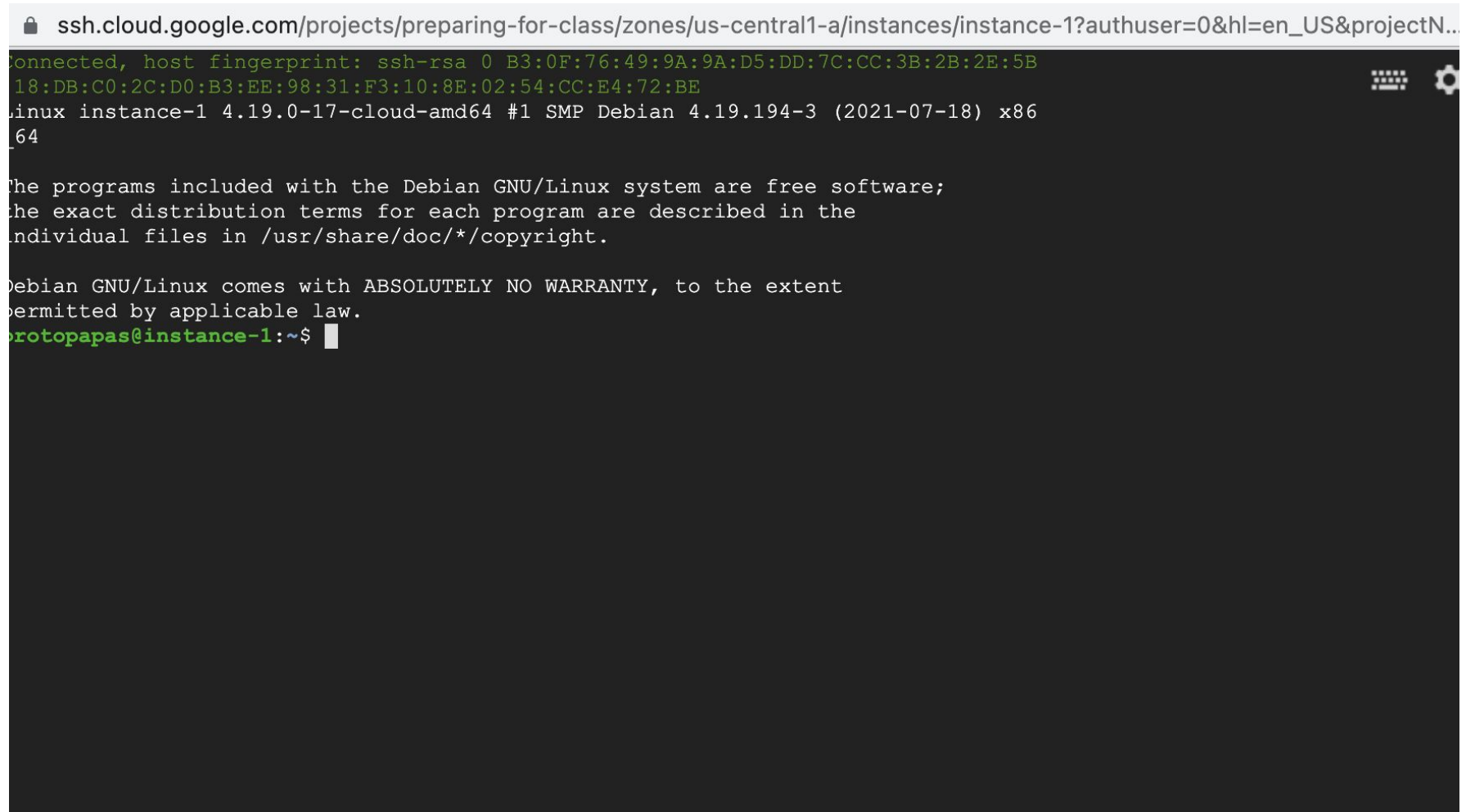
Related actions

DISMISS

- View billing report**
View and manage your Compute Engine billing
- Monitor VMs**
View outlier VMs across metrics like CPU and network
- Explore VM logs**
View, search, analyze, and download VM instance logs
- Set up firewall rules**
Control traffic to and from a VM instance
- Patch management**
Schedule patch updates and view patch compliance on VM instances

Virtual Machines Demo

And here is your virtual machine

A screenshot of a terminal window showing an SSH connection to a Google Cloud instance. The terminal has a dark background with light green text. The browser address bar at the top shows the URL: ssh.cloud.google.com/projects/preparing-for-class/zones/us-central1-a/instances/instance-1?authuser=0&hl=en_US&projectN... The terminal output shows the SSH connection process, including the host fingerprint and the system information of the remote machine: Linux instance-1 4.19.0-17-cloud-amd64 #1 SMP Debian 4.19.194-3 (2021-07-18) x86_64. It then displays the Debian GNU/Linux system's free software notice and warranty disclaimer. The prompt is now rootopapas@instance-1:~\$ with a cursor.

```
ssh.cloud.google.com/projects/preparing-for-class/zones/us-central1-a/instances/instance-1?authuser=0&hl=en_US&projectN...
Connected, host fingerprint: ssh-rsa 0 B3:0F:76:49:9A:9A:D5:DD:7C:CC:3B:2B:2E:5B
18:DB:C0:2C:D0:B3:EE:98:31:F3:10:8E:02:54:CC:E4:72:BE
Linux instance-1 4.19.0-17-cloud-amd64 #1 SMP Debian 4.19.194-3 (2021-07-18) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
rootopapas@instance-1:~$
```

Why should we use virtual machines?

Motivation

- All team members want an identical machine with same OS
- Should be easy create and delete instances
- All softwares installations need to be same across team members
- Team members need to run the same experiments in Isolation!

Virtual Machines!

Why should we use virtual machines?

Advantages

- Full autonomy: it works like a separate computer system; it is like running a computer within a computer.
- **Very secure**: the software inside the virtual machine cannot affect the actual computer.
- Lower costs: buy one machine and run multiple operating systems.
- Used by all Cloud providers for on demand server instances.

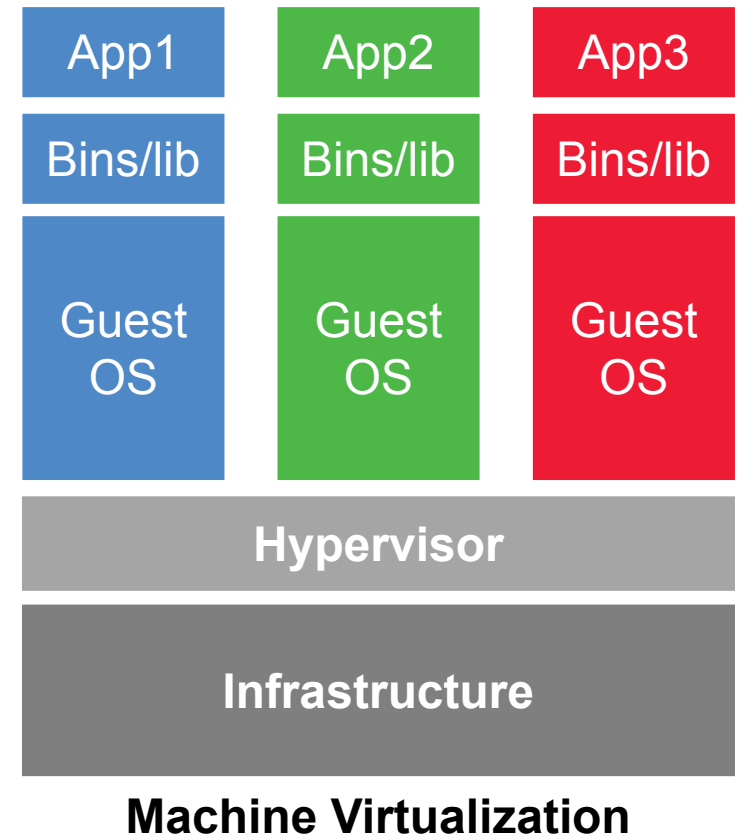
Softwares used for Virtualization

- VirtualBox
- VMWare
- Parallels

Why should we use virtual machines?

Advantages

- 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 we 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



Why should we use virtual machines?

Limitations

- Uses hardware in local machine
- Not very portable since size of VMs are large
- There is an overhead associated with virtual machines
 - Guest is not as fast as the host system
 - Takes a long time to start up
 - It may not have the same graphics capabilities

Outline

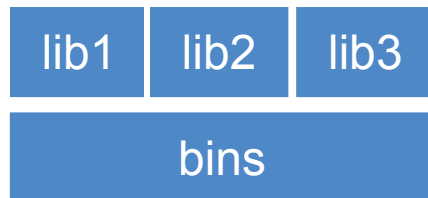
1. Where we are
2. Motivation
3. Virtual Machines
4. **Virtual Environments**

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 CS109, 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.



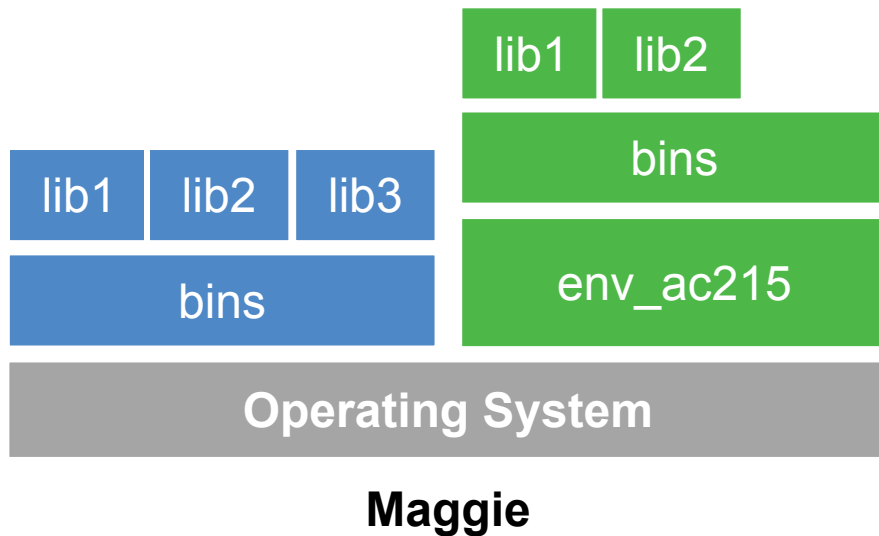
Operating System

Maggie

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

Why should we use virtual environment?

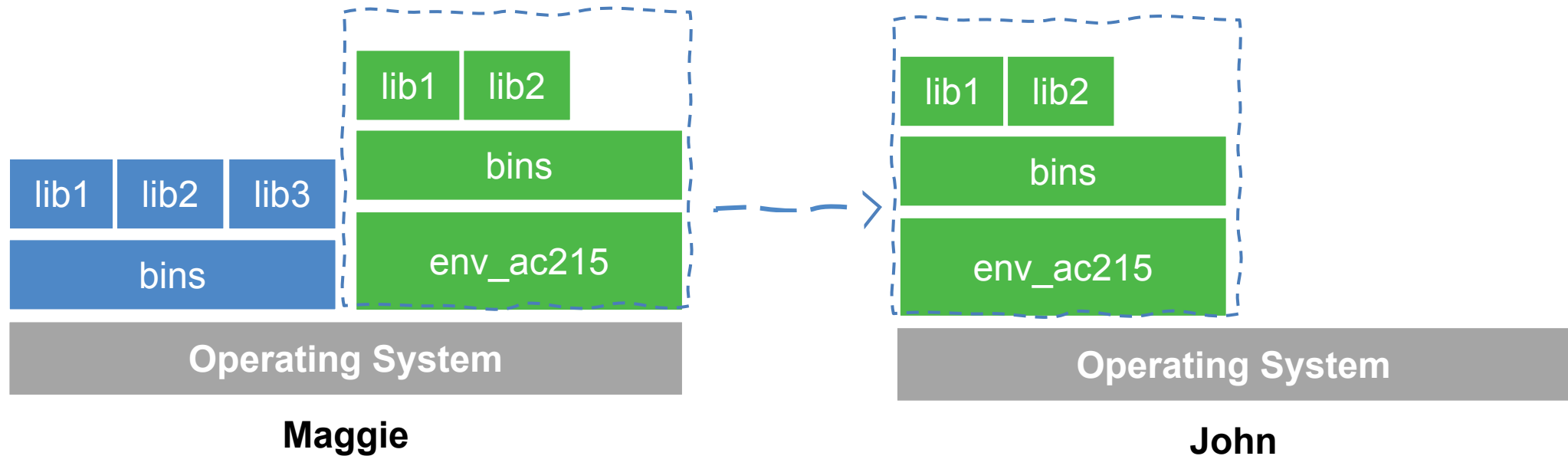
- Maggie starts taking AC215, and she thinks that it would be good to **isolate** the new environment from the previous environments avoiding any conflict with the installed packages. She adds a layer of **abstraction** called virtual environment that helps her keep the modules **organized** and avoid misbehaviors while developing a new project.



```
$ which python
/c/Users/maggie/Anaconda3/envs/env_ac215/python
```

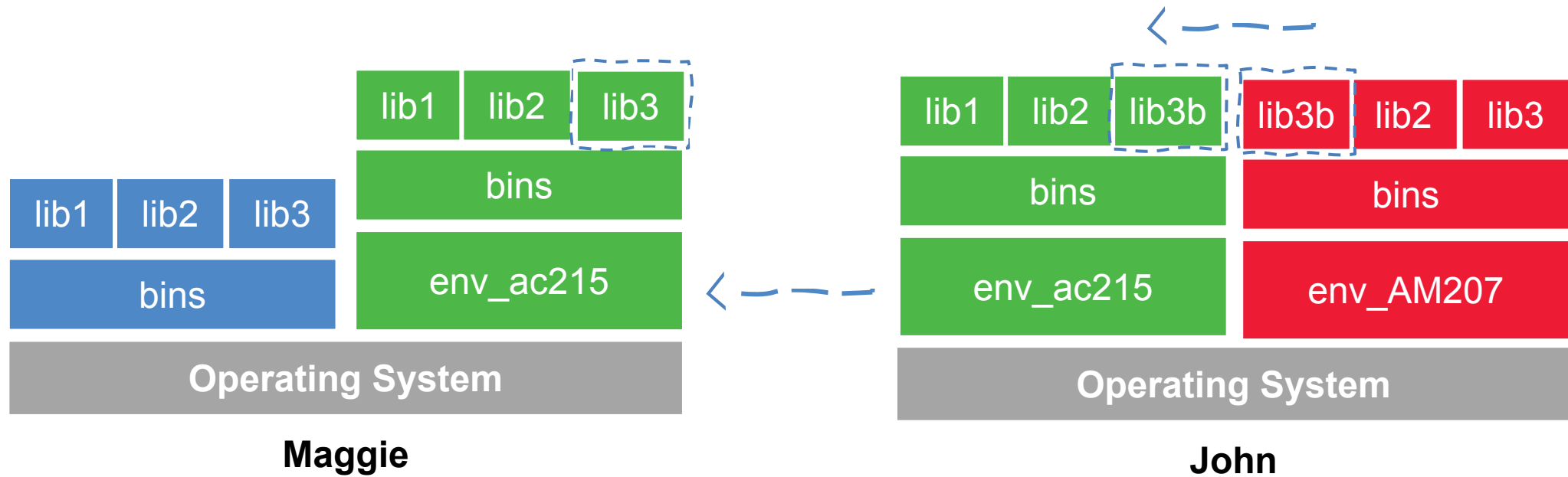
Why should we use virtual environment?

- Maggie collaborates with John for the final project and shares the environment she is working on through .yaml file (for conda env).



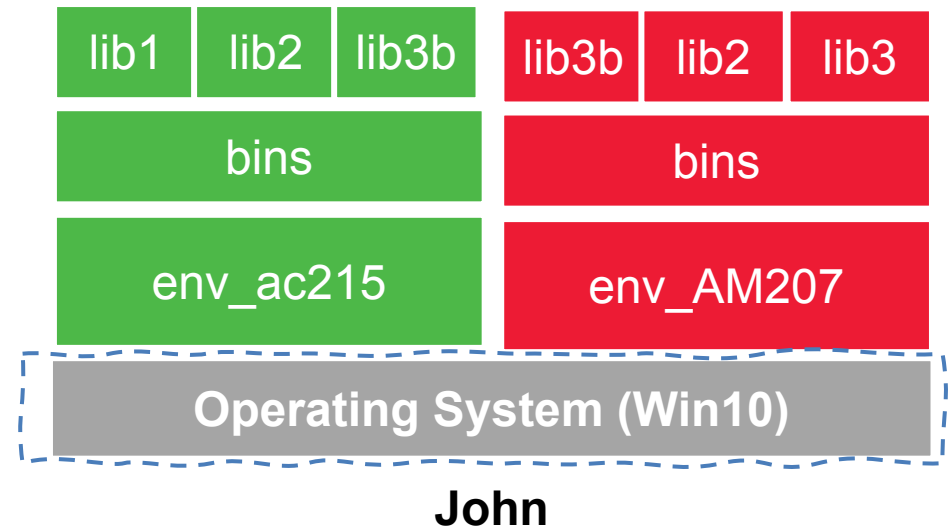
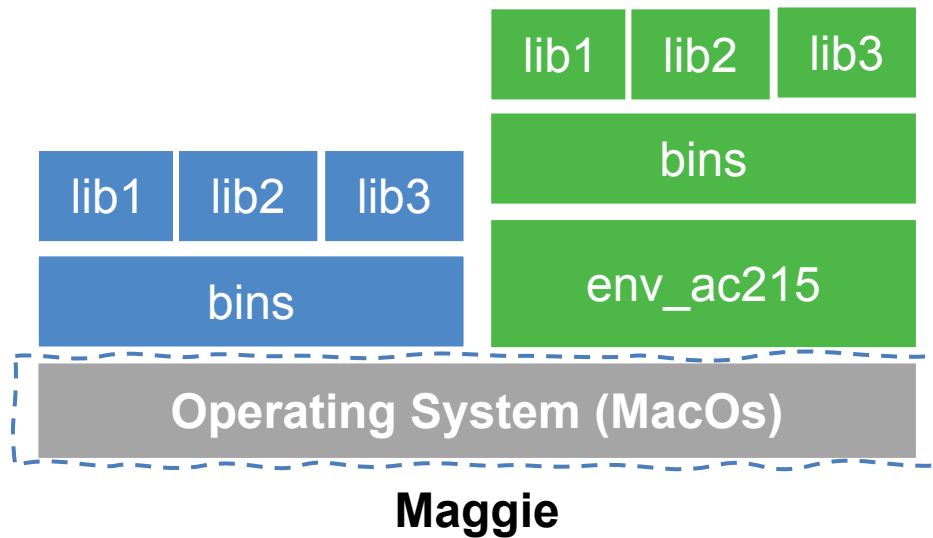
Why should we use virtual environment?

- John experiments with 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 .yaml file (for conda env). She can now update her environment and replicate the experiment.



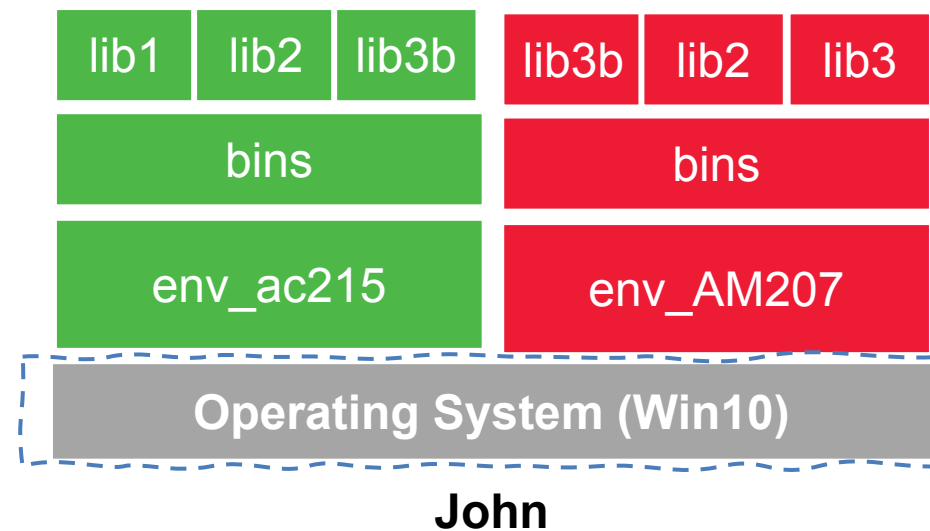
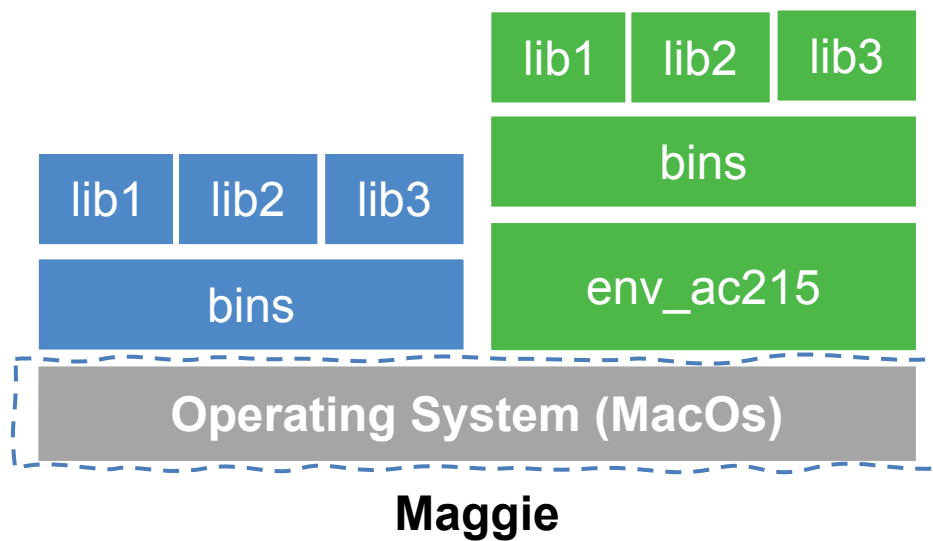
Why should we use virtual environment?

- 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.



Why should we use virtual environment?

- What could go wrong?



Virtual environments

Pros

- Reproducible research
- Explicit dependencies
- Improved engineering collaboration

Cons

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

What are virtual environments then?

- A virtual environment is an isolated python environment where the interpreter can run libraries and scripts independently from other virtual environments
- Think of a virtual environment as 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

Creating Virtual Environments

- **VirtualEnv**

The default way to create virtual environments in python

- **Conda**

Is a package manager and environment manager for Data Scientists

- **PipEnv**

Production-ready tool that aims to bring the best of all packaging worlds to the Python world

VirtualEnv

- Virtual environments manager embedded in Python
- Incorporated into broader tools such as pipenv
- Allow to install modules using pip package manager

VirtualEnv

How to use it:

- 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`

Conda

- Virtual environments manager embedded in Anaconda
- Allow to use both conda and pip to manage and install packages
- Virtual environments comes pre-installed with various engineering and data science packages

Conda

How to use it:

- 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`

Conda

How to use it:

- 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`

PipEnv

- Built on top of *VirtualEnv*
- Fixes many shortcomings of *VirtualEnv*
- Distinguish **development** vs. **production** environments
- Automatically keeps track of packages and package dependencies using a `Pipfile` & `Pipfile.lock`

PipEnv

How to use it:

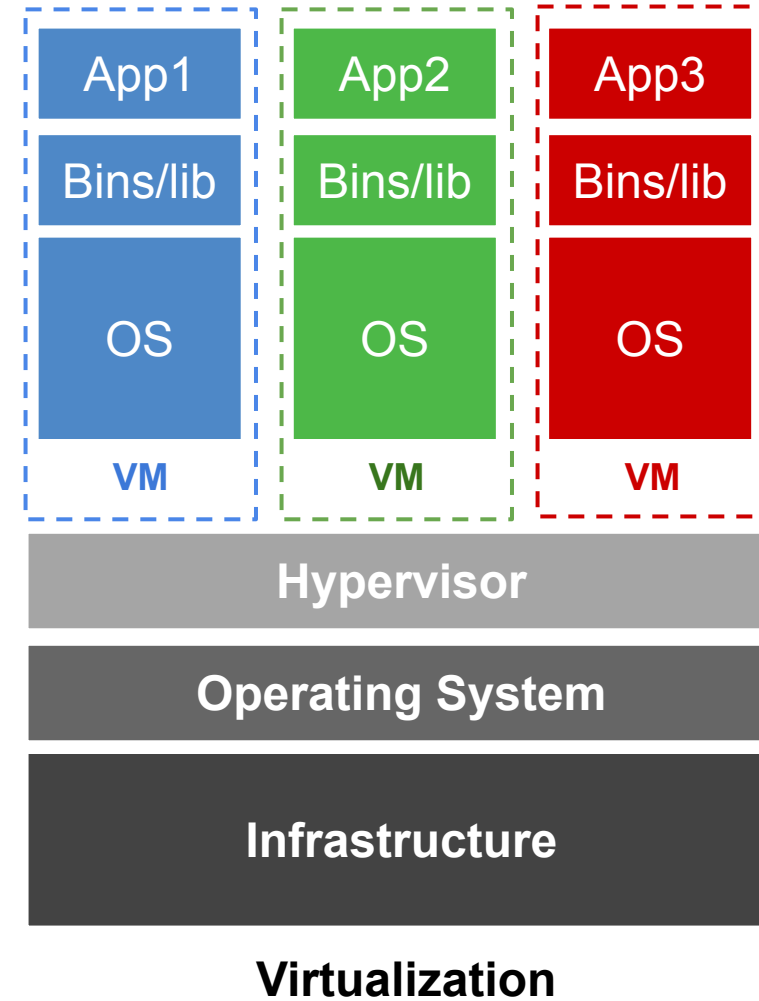
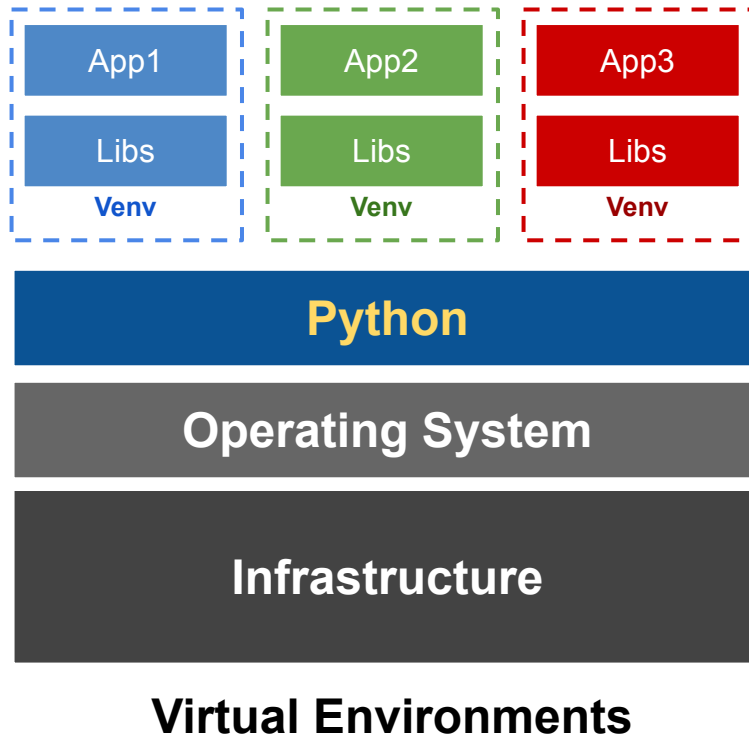
- Need to `pip install pipenv`
- To create a new environment run `pipenv install`
- Activate the environment by `pipenv shell`
- To install a new package `pipenv install numpy` or `pip install numpy` (this will not lock the package automatically)
- To sync from an existing Pipfile: `pipenv sync`

More on Virtual environments

Further readings

- Pipenv: Python Dev Workflow for Humans
<https://pipenv.pypa.io/en/latest/>
- 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>

Virtual Environments vs Virtual Machine



THANK YOU