

Lecture 2: Virtual Machines & Virtual Environments

AC215

Pavlos Protopapas
SEAS/Harvard



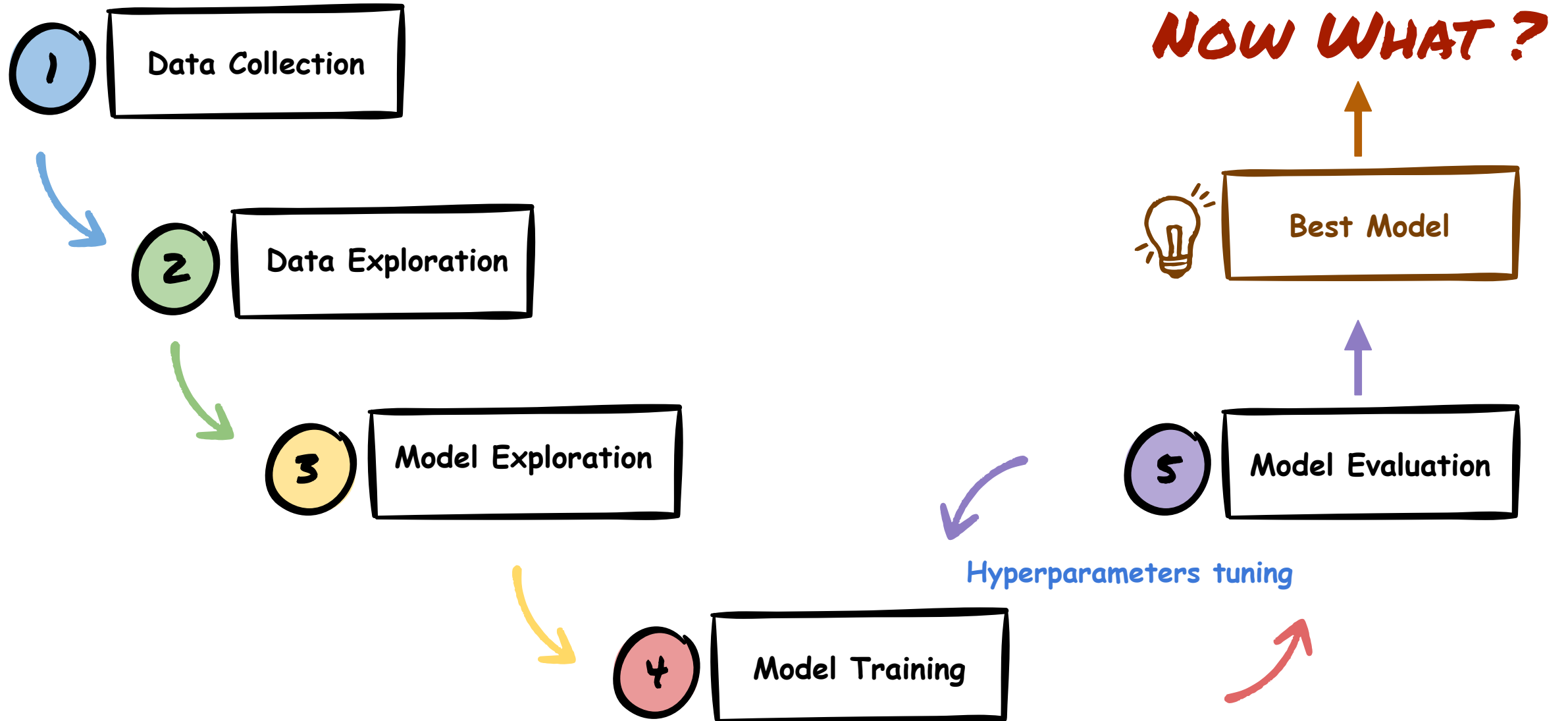
Outline

1. Motivation
2. Virtual Machines
3. Virtual Environments

Outline

- 1. Motivation**
2. Virtual Machines
3. Virtual Environments

Motivation: Deep Learning Flow

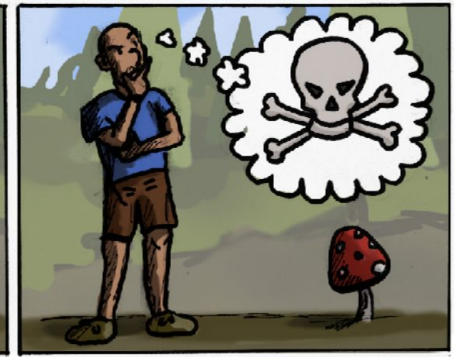


Motivation: Best Model

trainable_parameters	execution_time	loss	accuracy	model_size	learning_rate	batch_size	epochs	optimizer	name
2,306,051	2.97 mins	42.87	90.91%	10 MB	0.001	32	10	SGD	tfhub_mobilenetv2_train_base_True
82,179	3.19 mins	42.79	90.30%	10 MB	0.001	32	10	SGD	tfhub_mobilenetv2_train_base_False
164,355	3.91 mins	70.97	89.09%	10 MB	0.001	32	15	SGD	mobilenetv2_train_base_False
2,388,227	2.95 mins	82.03	88.48%	10 MB	0.001	32	10	SGD	mobilenetv2_train_base_True
11,112,323	6.85 mins	0.79	67.88%	44 MB	0.010	32	25	SGD	4_block
25,950,531	8.19 mins	0.74	66.67%	104 MB	0.010	32	25	SGD	2_block
22,514,755	4.78 mins	1.07	41.21%	90 MB	0.010	32	15	SGD	vgg_style

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

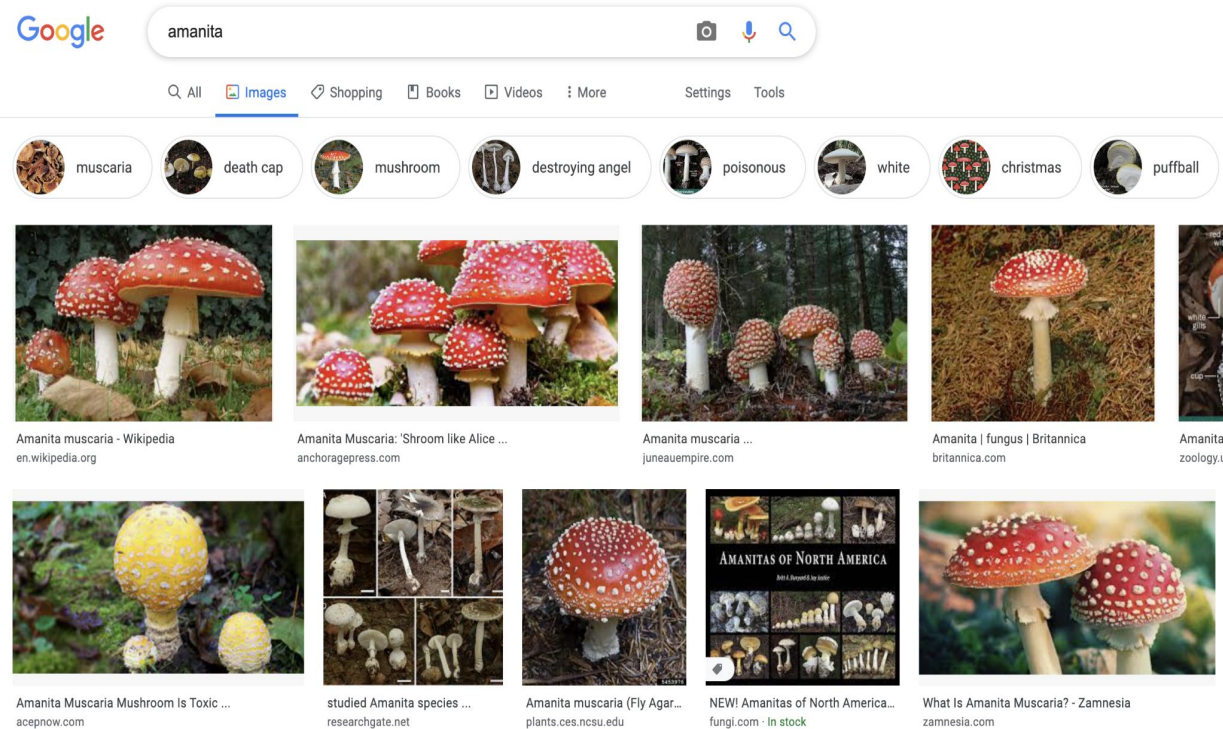


Credit: Nikolas Protopapas



Mushroom App: Data

- Collect images from Google
- For our demo we downloaded images for mushrooms **oyster**, **crimini**, **amanita (Poisonous)**
- Images organized into 3 labels



Python Script



Mushroom App: Models

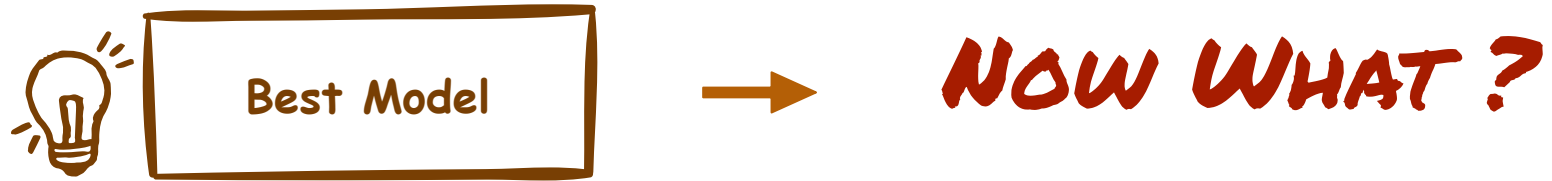
- Identify our problem task
- Try various model architectures
- Transfer Learning
- Hyperparameters tuning
- Experiment Tracking

trainable_parameters	execution_time	loss	accuracy
2,306,051	2.97 mins	42.87	90.91%
82,179	3.19 mins	42.79	90.30%
164,355	3.91 mins	70.97	89.09%
2,388,227	2.95 mins	82.03	88.48%
11,112,323	6.85 mins	0.79	67.88%
25,950,531	8.19 mins	0.74	66.67%
22,514,755	4.78 mins	1.07	41.21%

[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?



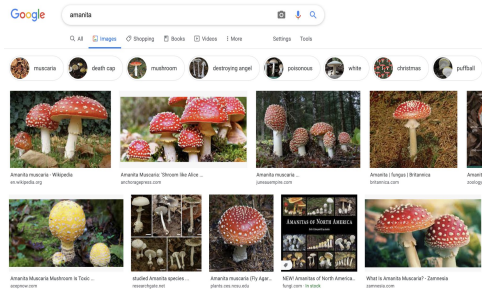
Type: amanita (93.54%)

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
- **Integrate** model prediction API into the app
- **Deploy** app to a cloud provider
- <http://awesome-mushroom-app.com> [Go live]

How do we build an App?

Data Collection



Python Script



O1_tutorial_mushroom_classification_models.ipynb

	trainable_parameters	execution_time	loss	accuracy	model_size
5	2,306,051	2.97 mins	42.87	90.91%	10 MB
4	82,179	3.19 mins	42.79	90.30%	10 MB
2	164,355	3.91 mins	70.97	89.09%	10 MB
6	2,388,227	2.95 mins	82.03	88.48%	10 MB
1	11,112,323	6.85 mins	0.79	67.88%	44 MB
0	25,950,531	8.19 mins	0.74	66.67%	104 MB
3	22,514,755	4.78 mins	1.07	41.21%	90 MB

Colab

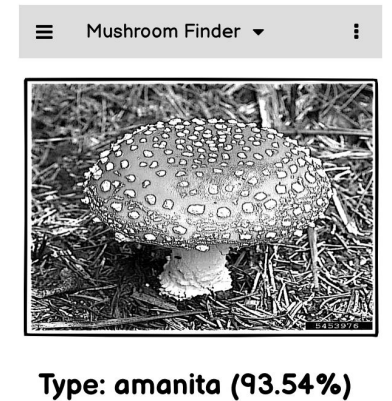
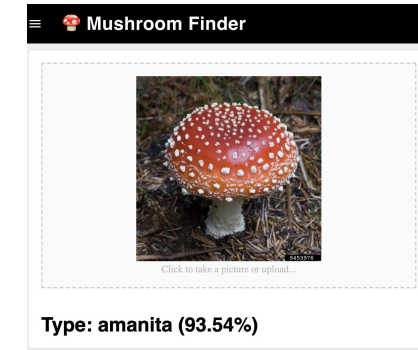


Rest API

Best Model

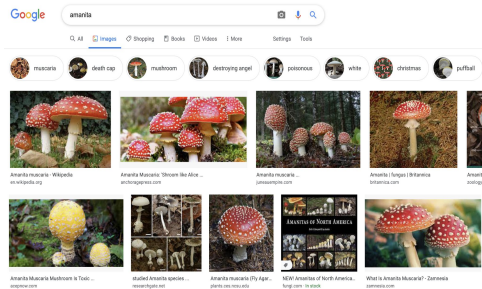
IDE / Code Editor

Data Exploration
Model Exploration
Model Training
Model Evaluation



How do we build an App?

Data Collection



Python Script

Data Exploration
Model Exploration
Model Training
Model Evaluation

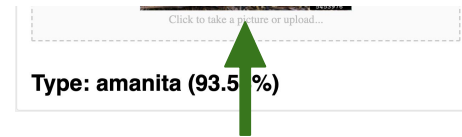
O1_tutorial_mushroom_classification_models.ipynb

	trainable_parameters	execution_time	loss	accuracy	model_size
5	2,306,051	2.97 mins	42.87	90.91%	10 MB
4	82,179	3.19 mins	42.79	90.30%	10 MB
2	164,355	3.91 mins	70.97	89.09%	10 MB
6	2,388,227	2.95 mins	82.03	88.48%	10 MB
1	11,112,323	6.85 mins	0.79	67.88%	44 MB
0	25,950,531	8.19 mins	0.74	66.67%	104 MB
3	22,514,755	4.78 mins	1.07	41.21%	90 MB

Colab

Mushroom Finder

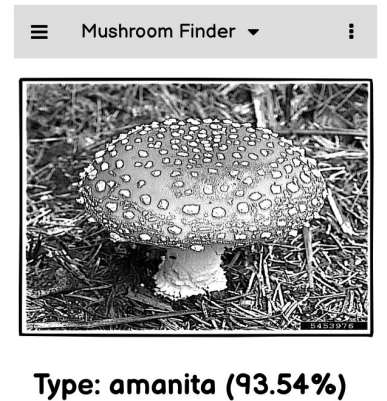
PRODUCTIONIZING MODEL !



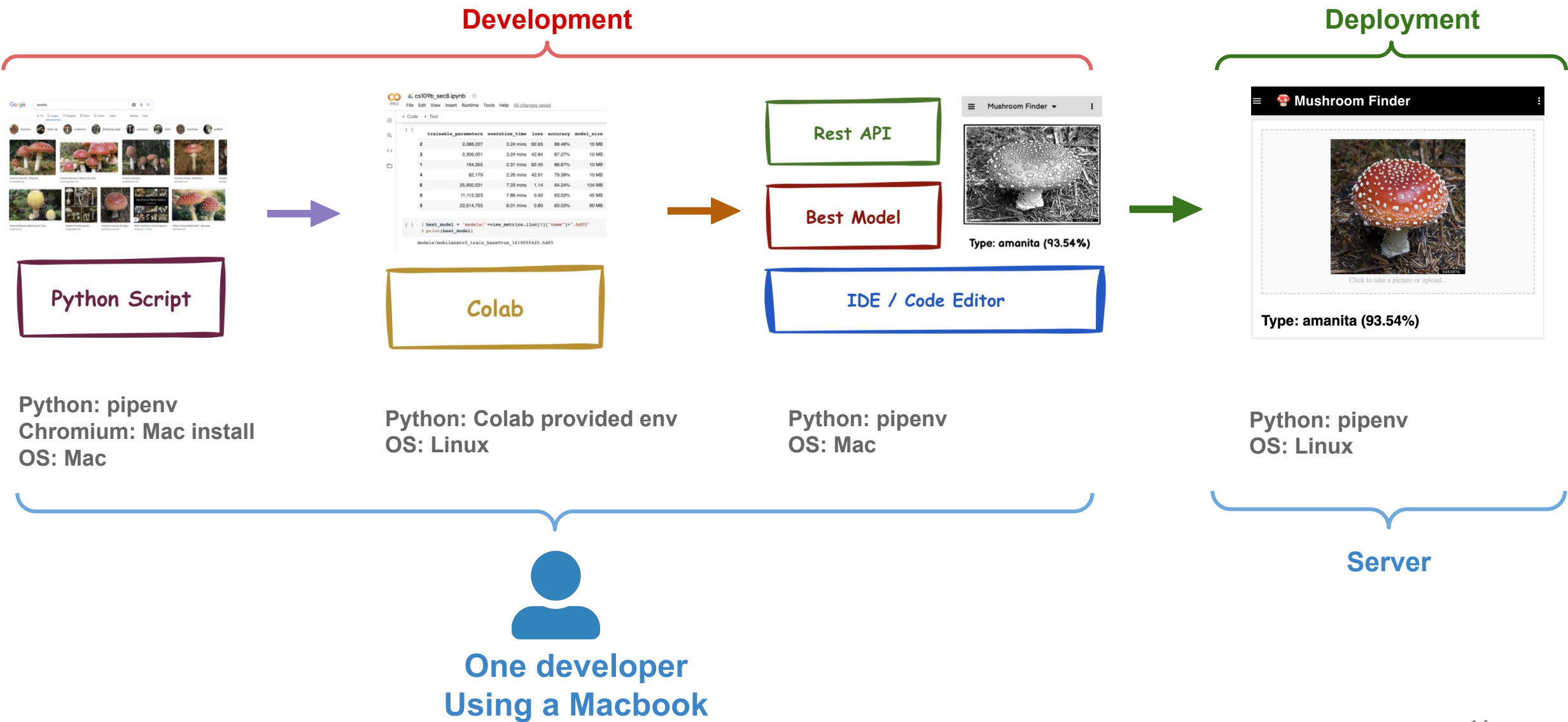
Rest API

Best Model

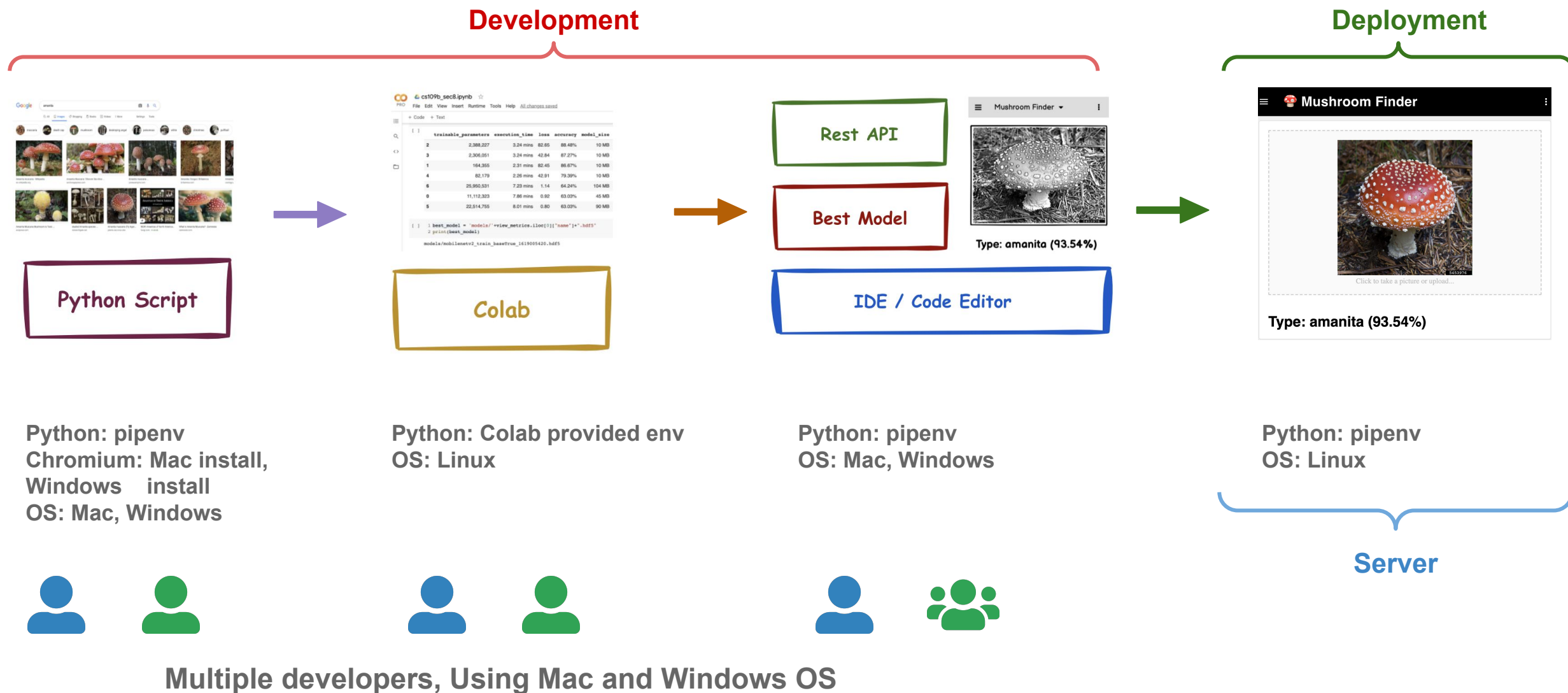
IDE / Code Editor



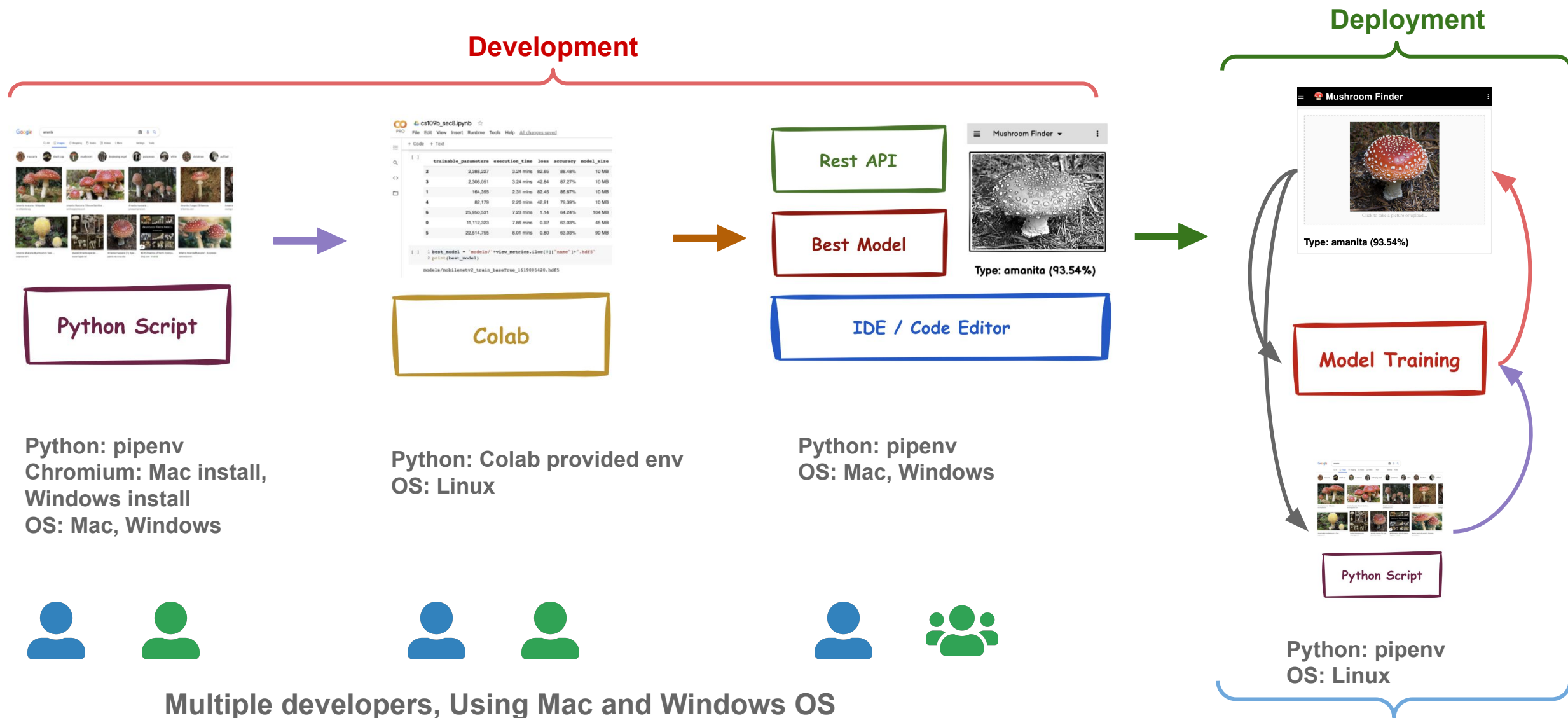
Challenges



Challenges - Multiple Developers



Challenges - Multiple Developers + Automation



Challenges / Solutions

Challenges:

- Required Installations for Specific Operating Systems
- Guidelines for Code Collaboration
- Methods for Sharing Datasets and Models
- Automation of Data Gathering and Model Training
- Onboarding Procedures for New Team Members
- Resolving "It Works on My Machine" Issues `_(ツ)_/`

Solutions:

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

Tools

- Virtual Machines
- Virtual Environments
- Containers

Outline

1. Motivation
- 2. Virtual Machines**
3. Virtual Environments

Virtual Machines Tutorial

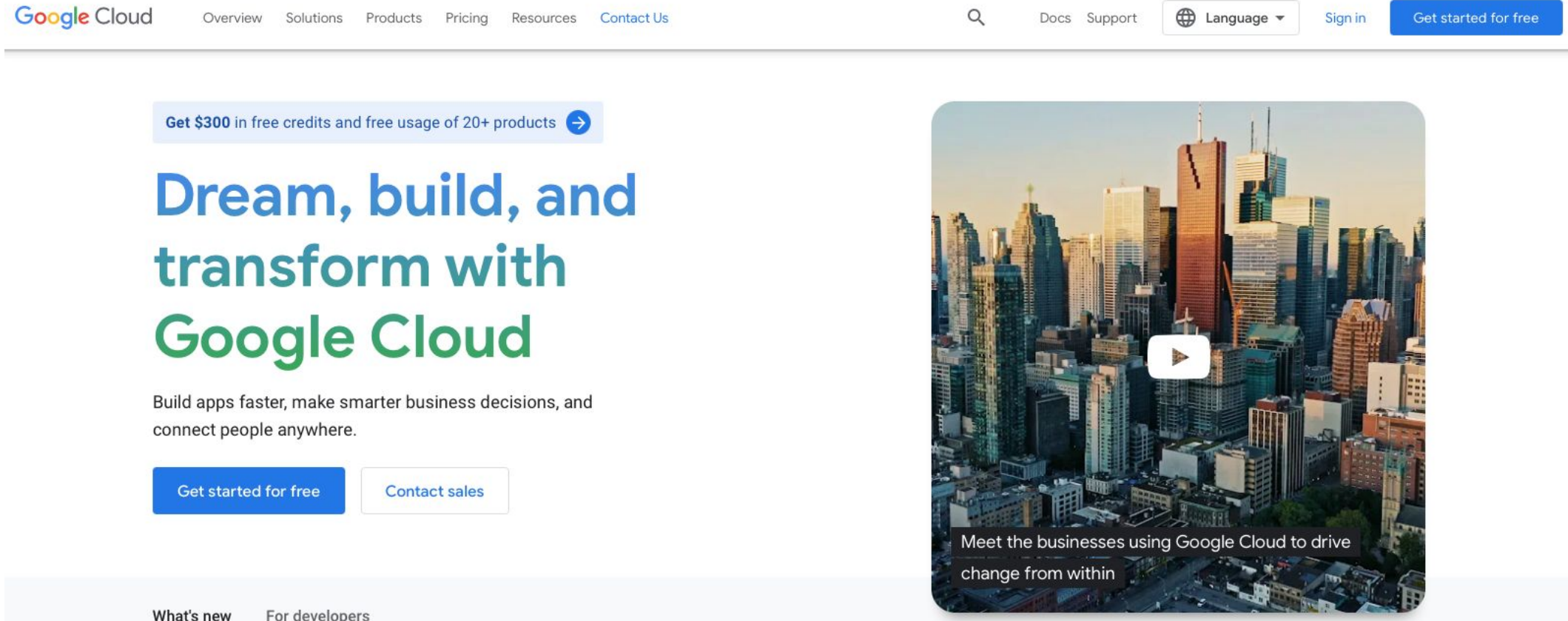
Running the Simple-Translate App on a **Virtual Machine**

To achieve this, follow the steps below:

- Create a Virtual Machine Instance.
- SSH into the Virtual Machine.
- Install Required Dependencies: git, Python.
- Download and Execute the Simple-Translate Python Script.
- For detailed instructions, please refer to the following link:
[Installing App on VM Manually](https://github.com/dlops-io/simple-translate#installing-app-on-vm-manually).
(<https://github.com/dlops-io/simple-translate#installing-app-on-vm-manually>)

Virtual Machines Tutorial

Google Cloud Platform: <https://cloud.google.com>



The screenshot shows the Google Cloud homepage. At the top is the Google Cloud logo and a navigation menu with links for Overview, Solutions, Products, Pricing, Resources, and Contact Us. On the right side of the header are search, Docs, Support, a language selector, a Sign in button, and a Get started for free button. The main content area features a promotional banner with the text 'Get \$300 in free credits and free usage of 20+ products' and a right arrow. Below this is the headline 'Dream, build, and transform with Google Cloud' in blue and green. Underneath the headline is the text 'Build apps faster, make smarter business decisions, and connect people anywhere.' and two buttons: 'Get started for free' and 'Contact sales'. To the right of the text is a large video player showing a city skyline at sunset with a play button overlay. At the bottom of the video player is the text 'Meet the businesses using Google Cloud to drive change from within'. The footer contains links for 'What's new' and 'For developers'.

Google Cloud

Overview Solutions Products Pricing Resources Contact Us

Search Docs Support Language Sign in Get started for free

Get \$300 in free credits and free usage of 20+ products →

Dream, build, and transform with Google Cloud

Build apps faster, make smarter business decisions, and connect people anywhere.

Get started for free Contact sales

Meet the businesses using Google Cloud to drive change from within

What's new For developers

Virtual Machines Tutorial

Go to Navigation Menu

Google Cloud Platform

Preparing For Class

Search products and resources

Navigation menu

ACTIVITY

RECOMMENDATIONS

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

No data is available for the selected time frame.

Google Cloud Platform status

All services normal

[Go to Cloud status dashboard](#)

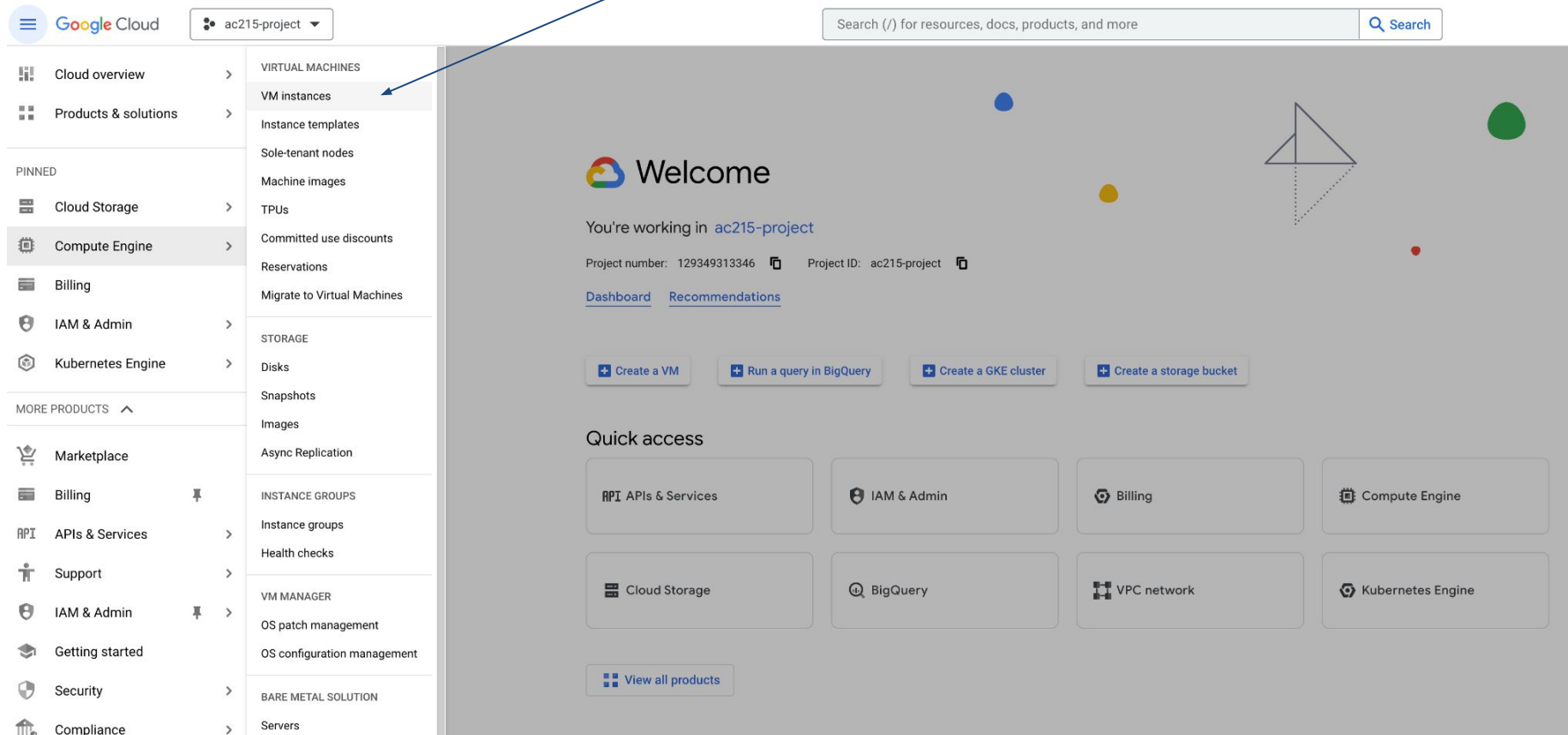
Billing

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

USD \$0.00

Virtual Machines Tutorial

Select compute engine



The screenshot displays the Google Cloud console interface for the project 'ac215-project'. The left-hand navigation menu is expanded, showing various service categories. Under the 'VIRTUAL MACHINES' section, the 'VM instances' option is highlighted, and a blue arrow points to it from the text 'Select compute engine' above. Other options in the menu include 'Instance templates', 'Sole-tenant nodes', 'Machine images', 'TPUs', 'Committed use discounts', 'Reservations', and 'Migrate to Virtual Machines'. The main dashboard area shows a 'Welcome' message, project details (Project number: 129349313346, Project ID: ac215-project), and quick access buttons for 'Create a VM', 'Run a query in BigQuery', 'Create a GKE cluster', and 'Create a storage bucket'. Below this, a 'Quick access' section provides shortcuts to 'APIs & Services', 'IAM & Admin', 'Billing', 'Compute Engine', 'Cloud Storage', 'BigQuery', 'VPC network', and 'Kubernetes Engine'. A 'View all products' button is also visible at the bottom.

Virtual Machines Tutorial

Select Virtual Machines

The screenshot displays the Google Cloud Platform console interface. At the top, the navigation bar includes the Google Cloud Platform logo, a dropdown menu for 'Preparing For Class', and a search bar. Below this, the 'Compute Engine' section is active, showing 'VM instances' as the selected resource. The left sidebar contains a navigation menu with categories like 'Virtual machines', 'Storage', 'Instance groups', 'VM Manager', and 'Settings'. The main content area features a 'Filter' input field and a table with columns for 'Status', 'Name', 'Zone', 'Recommendations', 'In use by', 'Internal IP', 'External IP', and 'Connect'. Below the table, there is a large graphic with the text 'VM Instances' and a description: '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.' At the bottom of this section are two buttons: 'CREATE INSTANCE' and 'TAKE THE QUICKSTART'.

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

Filter Enter property name or value

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

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 Tutorial

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 Tutorial

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

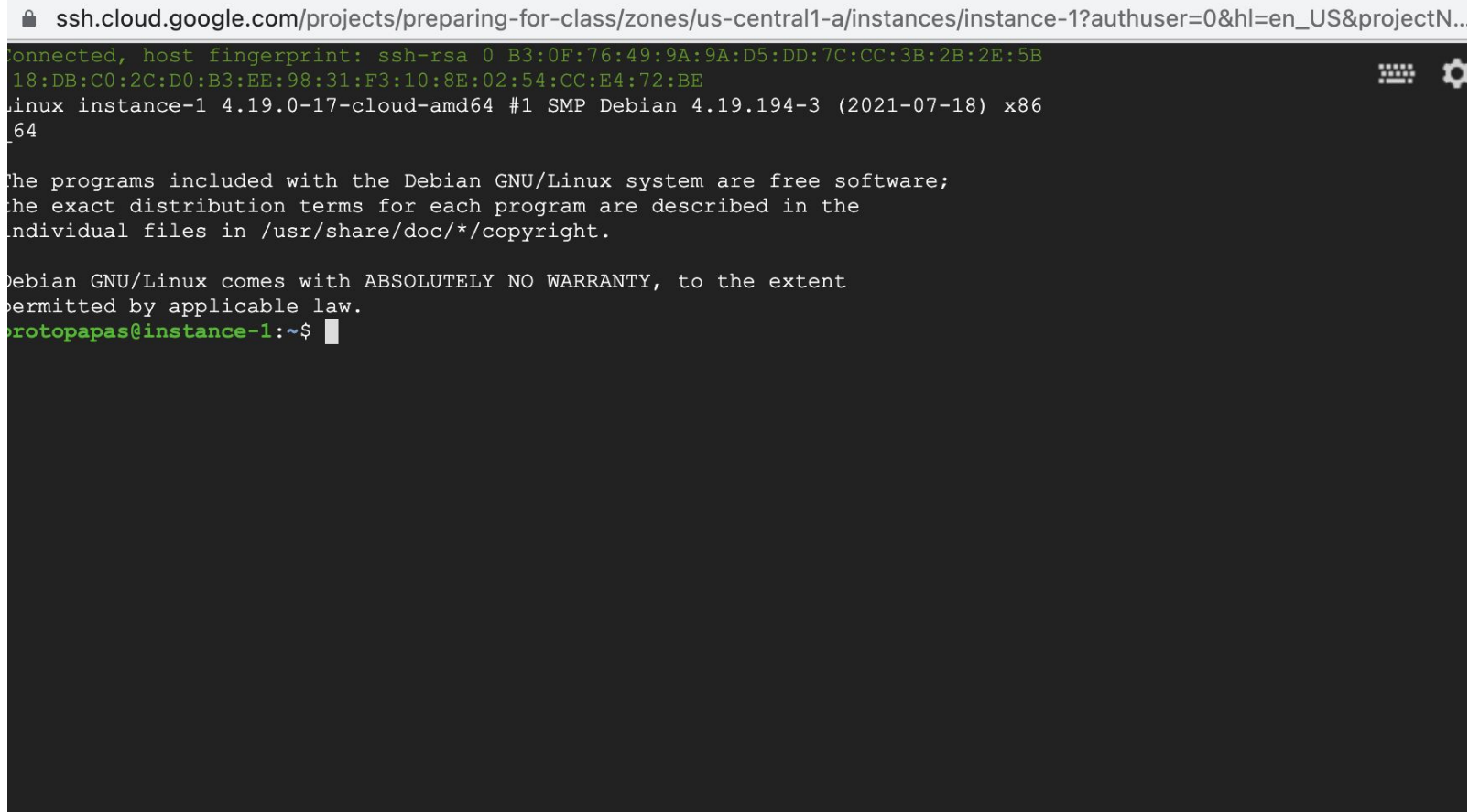
Related actions

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 Tutorial

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 and white text. The top line shows the URL: ssh.cloud.google.com/projects/preparing-for-class/zones/us-central1-a/instances/instance-1?authuser=0&hl=en_US&projectN... Below this, the terminal displays the SSH connection details, including the host fingerprint and the operating system information: Linux instance-1 4.19.0-17-cloud-amd64 #1 SMP Debian 4.19.194-3 (2021-07-18) x86_64. The terminal then shows the Debian GNU/Linux system's free software notice and the user's prompt: rootopapas@instance-1:~\$.

```
git clone https://github.com/dlops-io/simple-translate.git
```

Why should we use virtual machines?

Motivation

- **Uniform Operating Environments:** Desire for a standardized OS across all team member workstations.
- **Consistent Software Configuration:** Requirement for identical software setups across the team.
- **Effortless Instance Management:** The need for simple procedures to instantiate and terminate VMs.
- **Consistent Software Configuration:** Requirement for identical software setups across the team.

Virtual Machines!

Why should we use virtual machines?

Advantages

- **Complete Autonomy:** it works like a separate computer system; it is like running a computer within a computer.
- **Enhance Security:** the software inside the virtual machine cannot affect the actual computer.
- **Cost-Effectiveness:** Purchase a single machine and run multiple operating systems.
- **Widely Adopted:** Utilized by all major cloud providers for on-demand server instances.

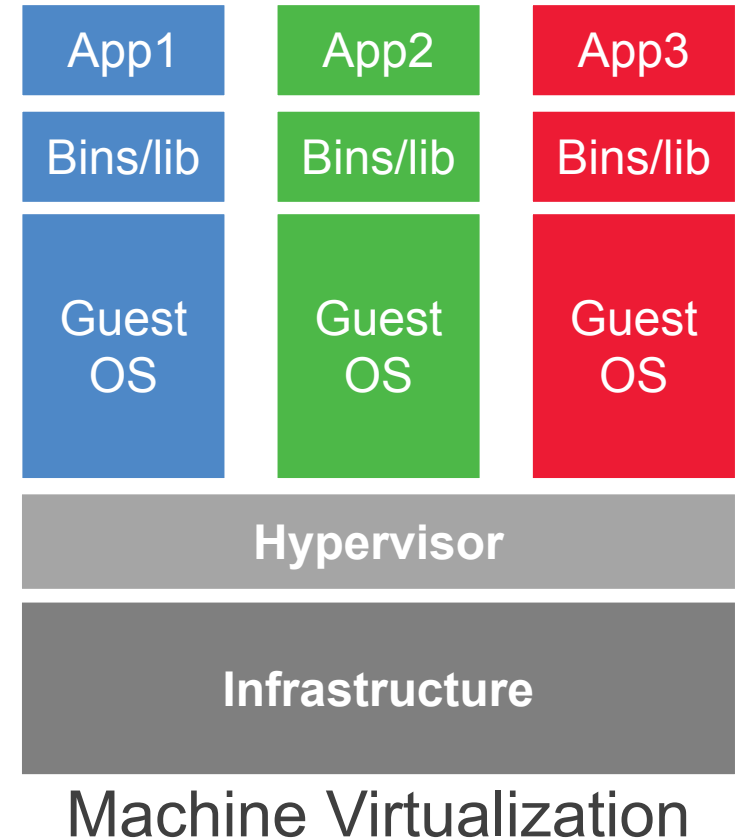
Software for Virtualization

- 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

- **Local Hardware Dependency:** Relies on the hardware resources of the host machine.
- **Limited Portability:** Large file sizes can impede easy transfer or deployment.
- **Resource Overhead:** Additional computational and memory resources are required to operate.
- **Reduced Performance:** The guest system typically runs slower than the host environment.
- **Slow Initialization:** Extended startup times compared to native systems.
- **Graphics Constraints:** May lack the graphical capabilities of the host system.

Outline

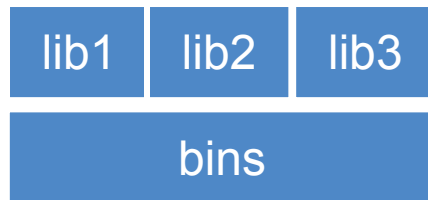
1. Motivation
2. Virtual Machines
3. **Virtual Environments**

Why should we use virtual environment?

- **Streamlines** code development and usage.
- Isolates dependencies in separate "**sandboxes**" for easy switching between applications.
- Given an operating system and hardware, we can get the exact code environment set up using **different technologies**.

Why should we use virtual environment?

Maggie took CS109B and used to run her Jupyter notebooks from the Anaconda prompt. Whenever she installed a module, it was placed in one of the following folders: `bin`, `lib`, `share`, or `include`. She could then import the module and used it without any issue.



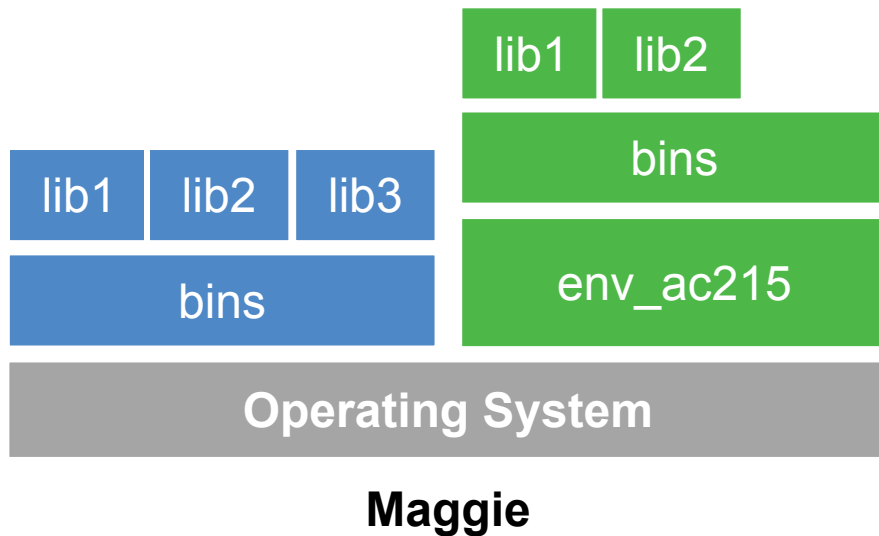
Operating System

Maggie

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

Why should we use virtual environment?

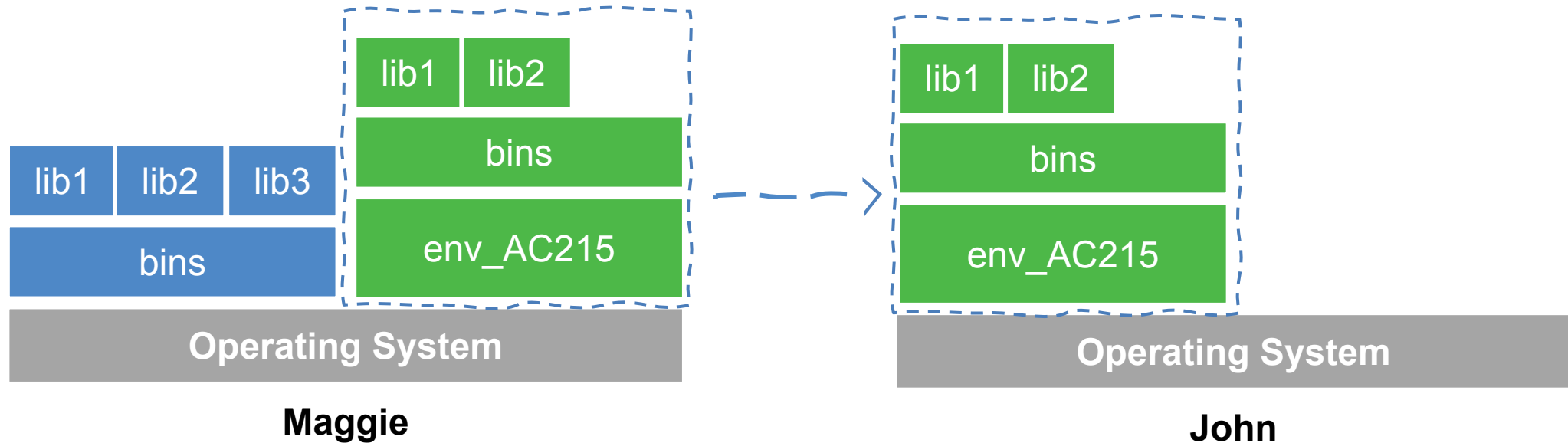
Maggie begins taking AC215 and decides that **isolating** the new coding environment from previous ones would be beneficial to avoid package conflicts. To achieve this, she employs a layer of **abstraction** known as a virtual environment. This helps her keep modules organized and prevents issues while developing new projects.



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

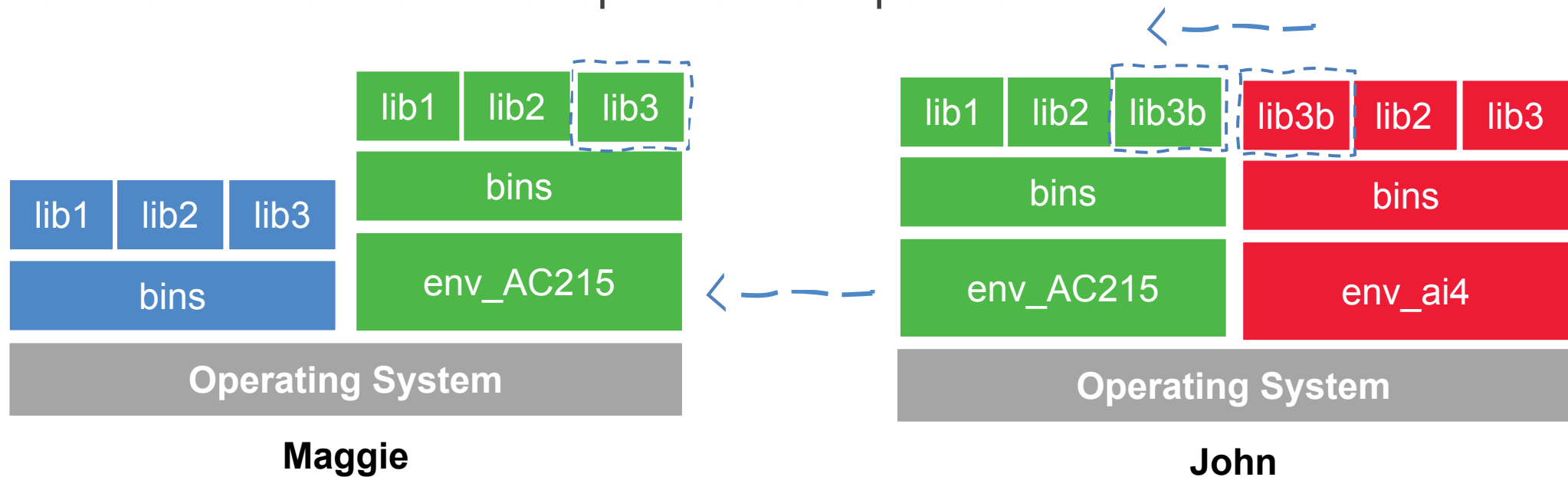
Why should we use virtual environment?

For the final project, Maggie collaborates with John and shares her working environment by distributing a .yml file for the Conda environment.



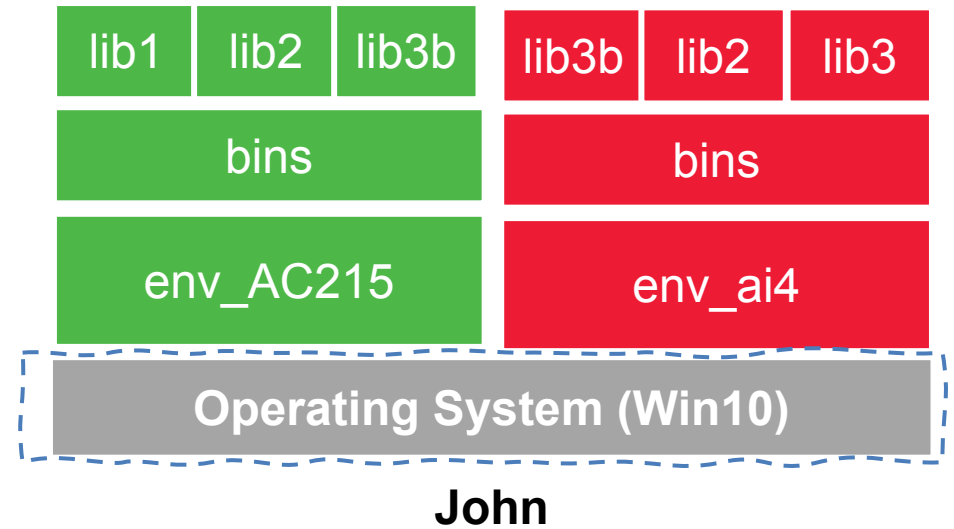
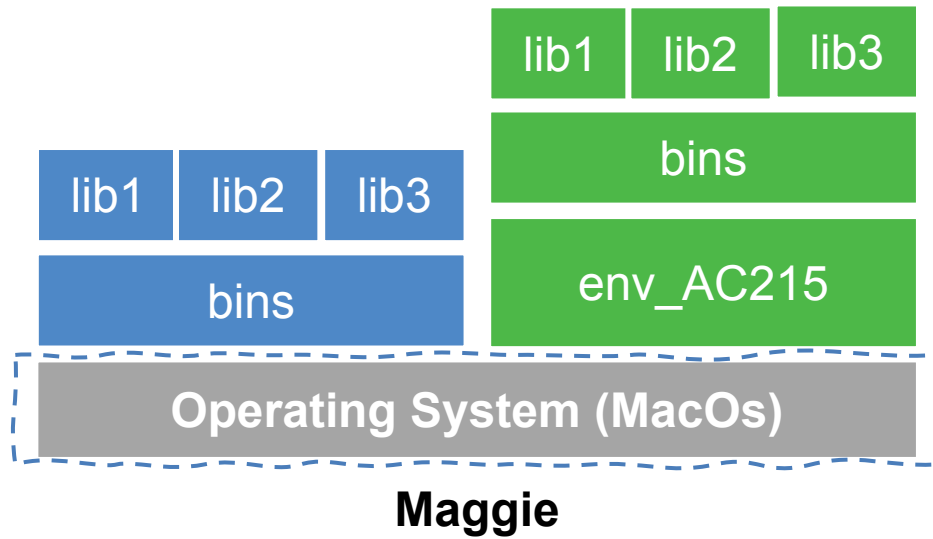
Why should we use virtual environment?

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.



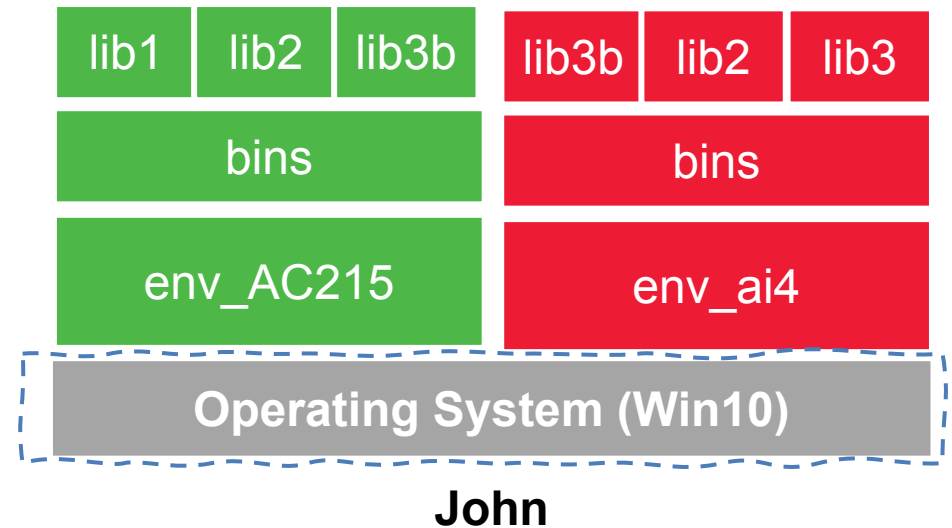
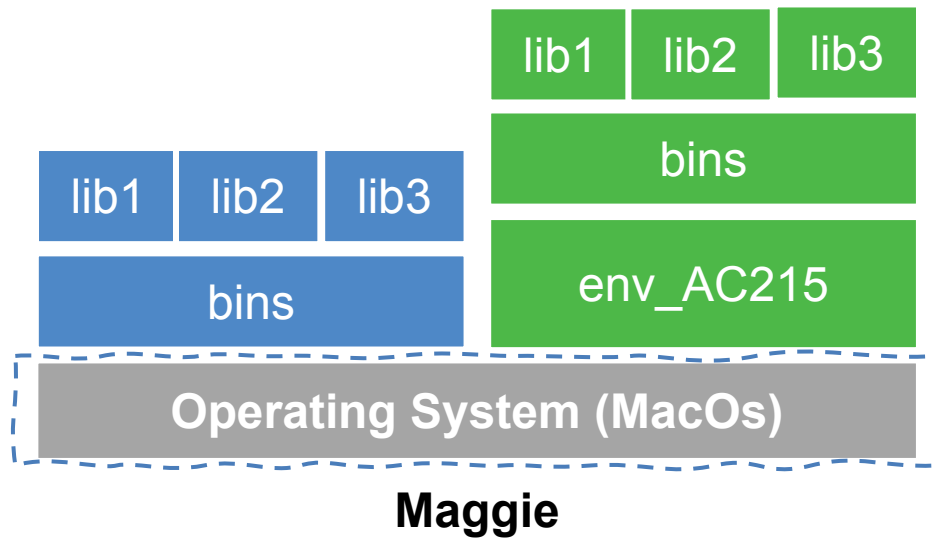
Why should we use virtual environment?

- What could go wrong?



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.



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`

- 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`

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

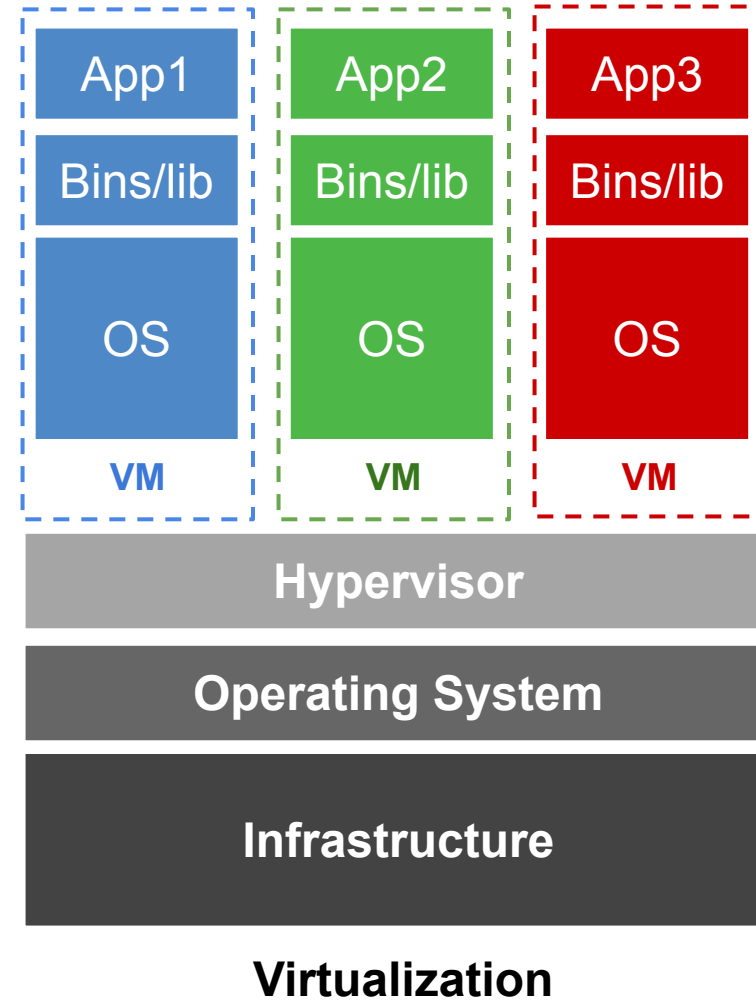
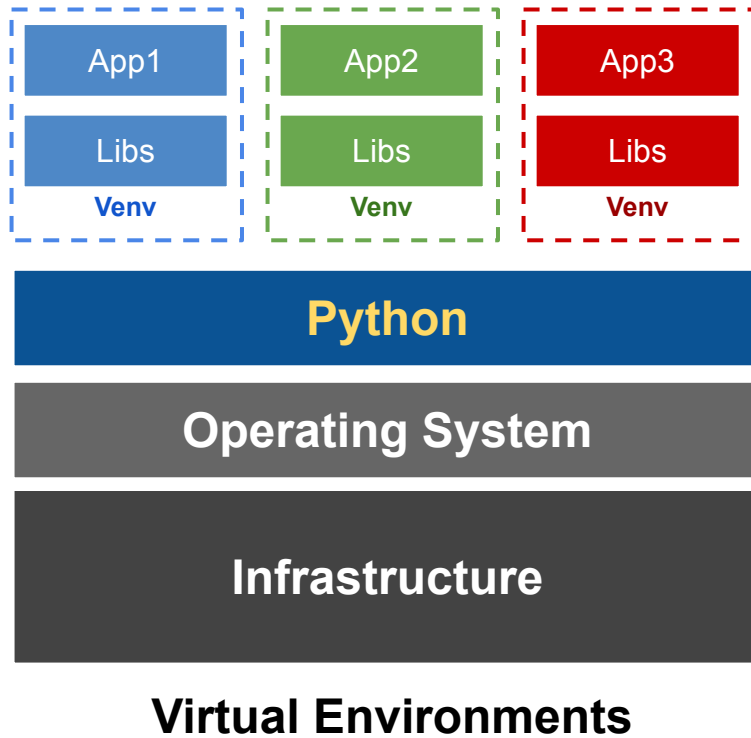
<https://jakevdp.github.io/blog/2016/08/25/conda-myths-and-misconceptions/>

- 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



Virtual Environment Tutorial

- Let us run the simple-translate app using Virtual Environment
- For this we will do the following:
 - Create a VM Instance
 - SSH into the VM
 - Install dependencies: git, python
 - Download and run the simple-translate python script
- Full instructions can be found [here](#)

THANK YOU