UAL Workshops: Using a local server for research and work

Matias Quintana
2023-04-17

Housekeeping

Housekeeping

Housekeeping

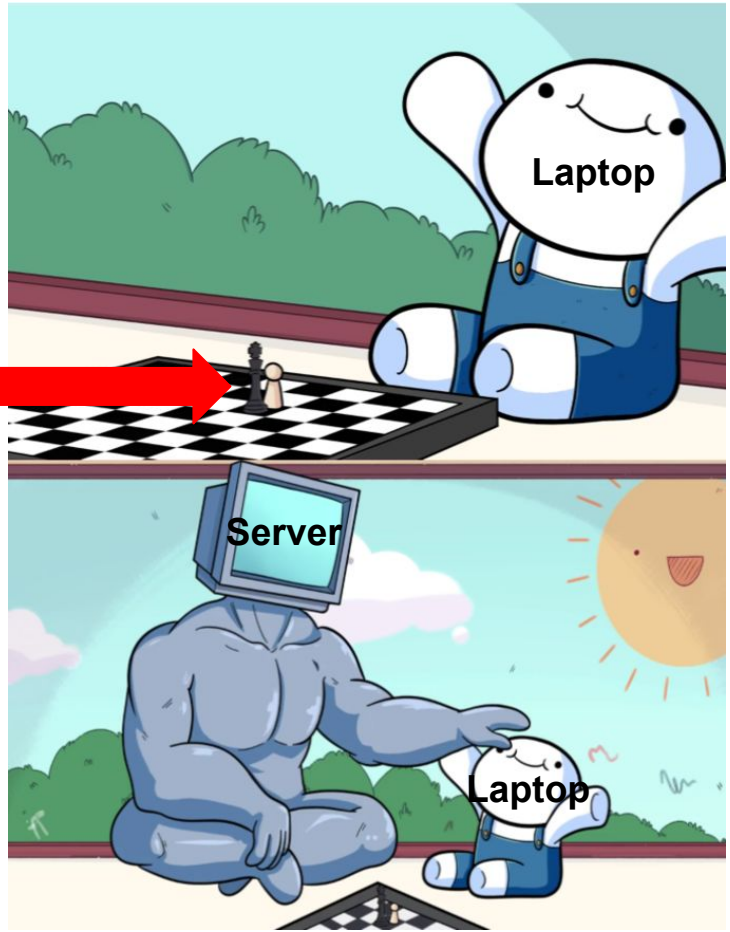
If something doesn’t make sense or is not clear,

PLEASE ASK!

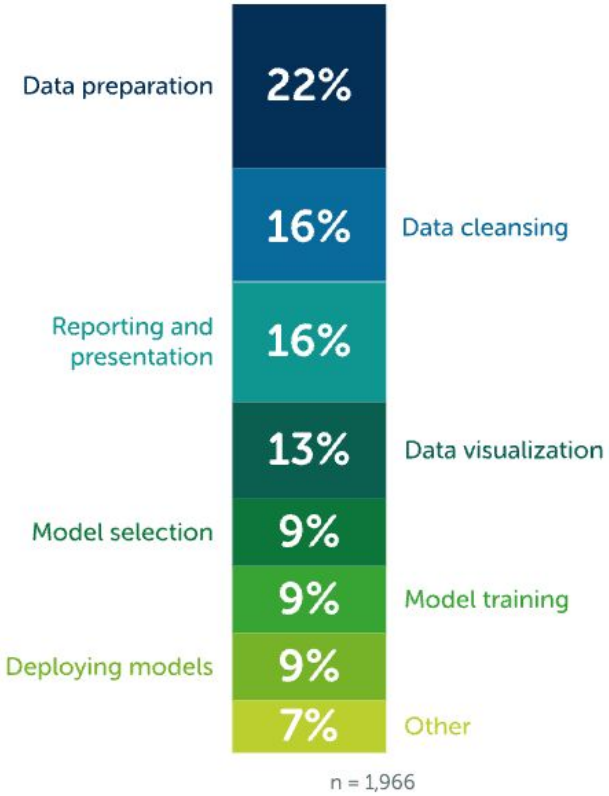
Why?

Why?

Running ML models

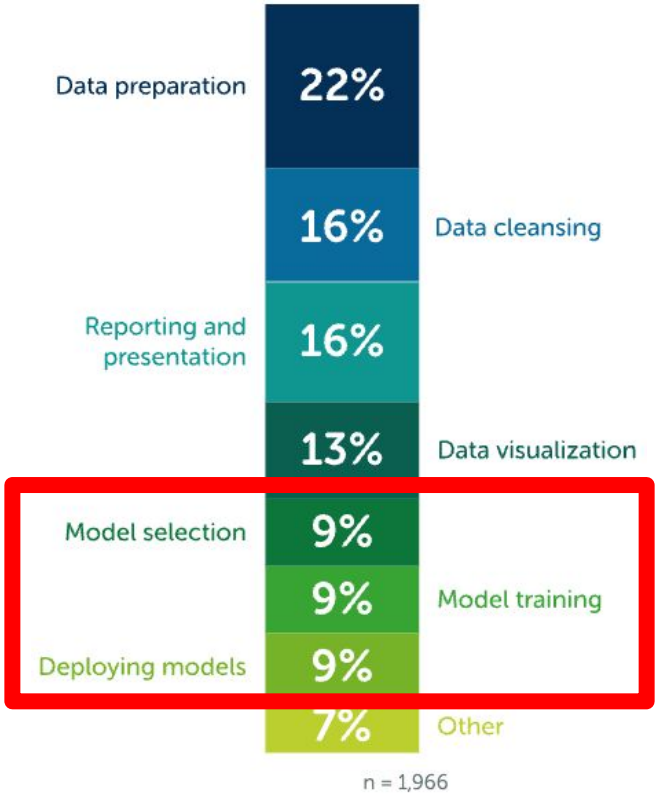


Why?



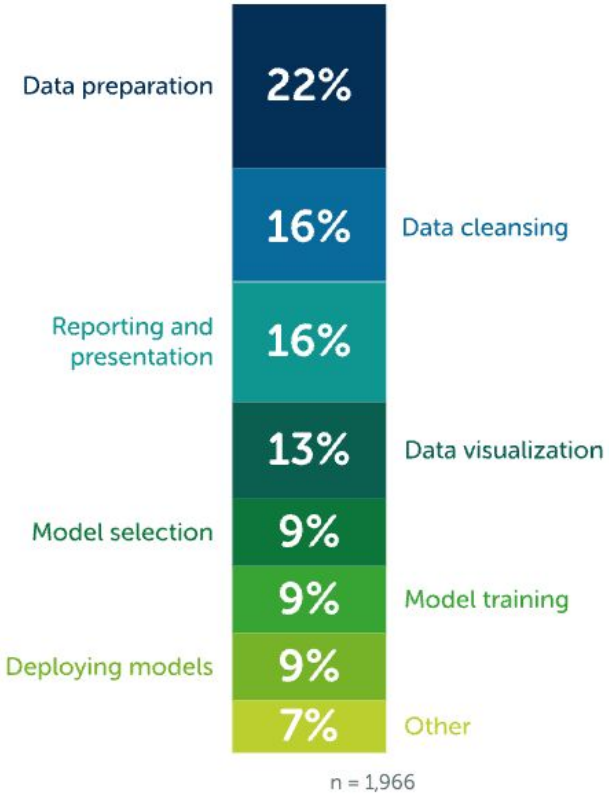
https://www.anaconda.com/state-of-data-science-report-2022

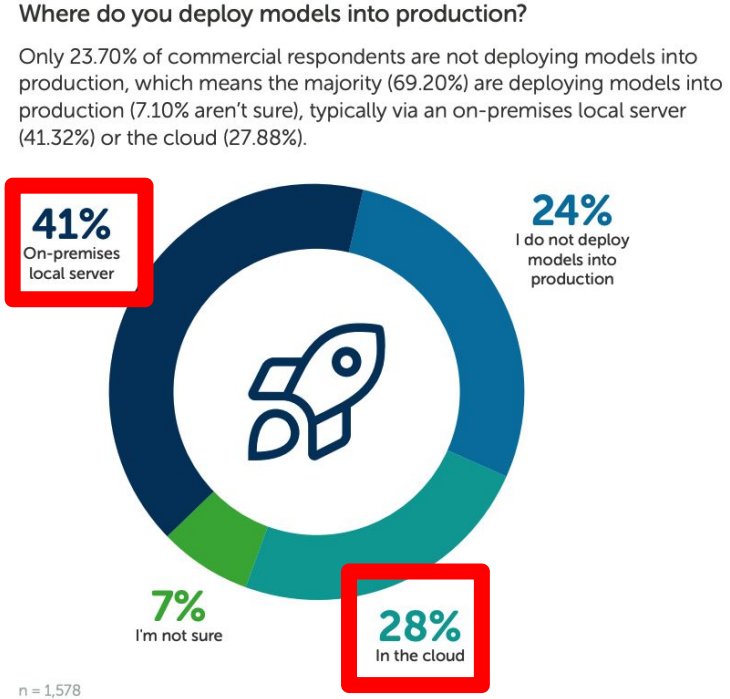
Why?



https://www.anaconda.com/state-of-data-science-report-2022

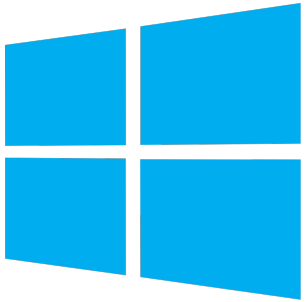
Why?



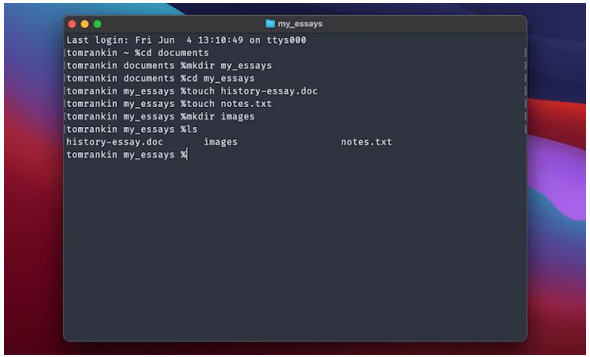


https://www.anaconda.com/state-of-data-science-report-2022

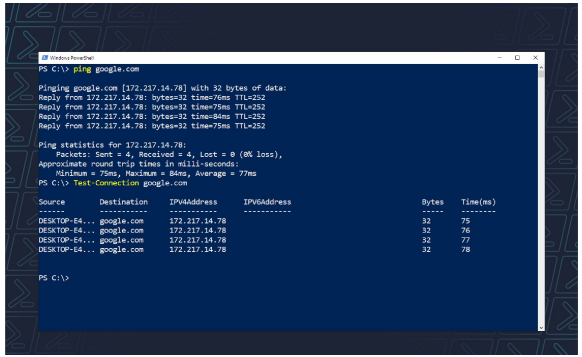
How to connect?



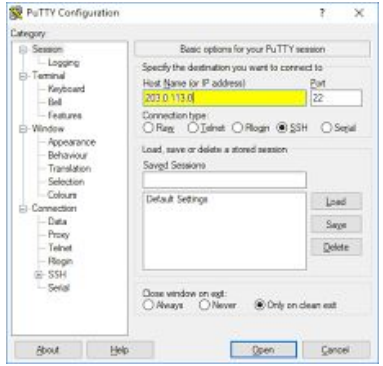
Terminal



Powershell

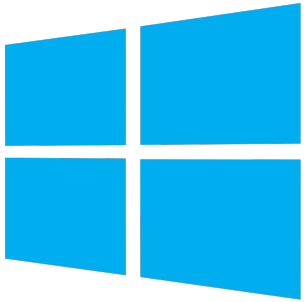


PuTTY

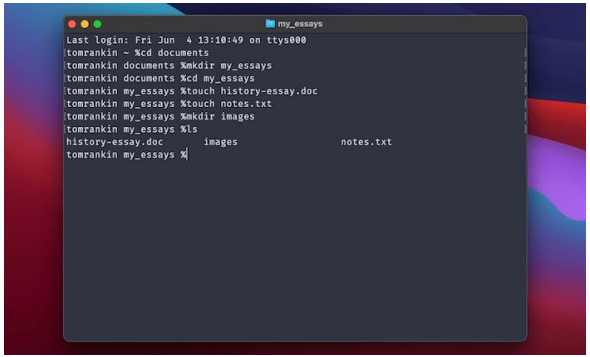


https://www.putty.org/

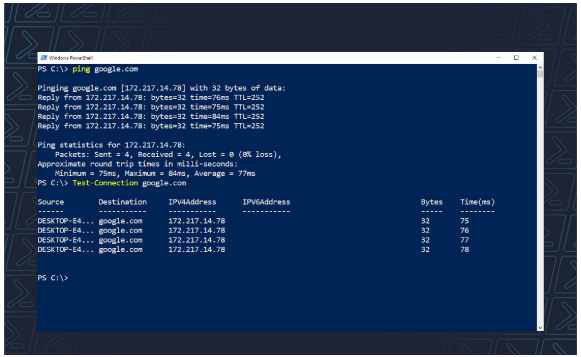
How to connect?



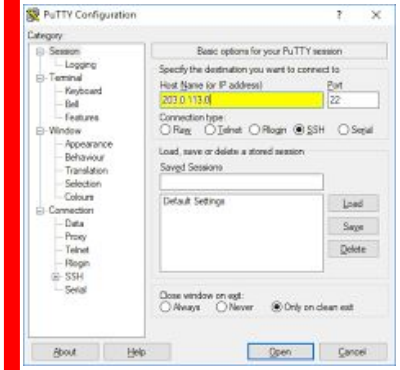
Terminal



Powershell



PuTTY



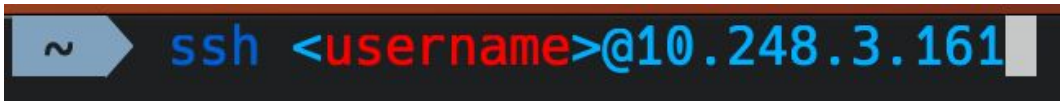
https://www.putty.org/

How to connect?

IP: **10.248.3.161**

Username: **your_first_name**

Password: **ual2023<number_of_characters_in_your_first_name>**

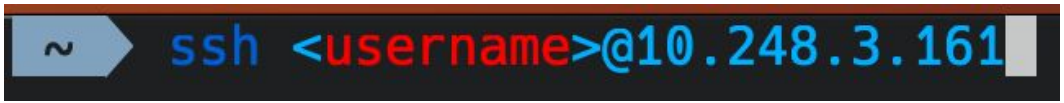


How to connect?

IP: **10.248.3.161**

Username: **your_first_name**

Password: **ual2023<number_of_characters_in_your_first_name>**



PLEASE change your password using **passwd**

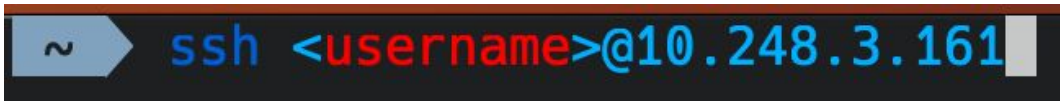
How to connect?

You can only connect within the NUS network.
Off Campus? **VPN** to NUS, then connect to server

IP: **10.248.3.161**

Username: **your_first_name**

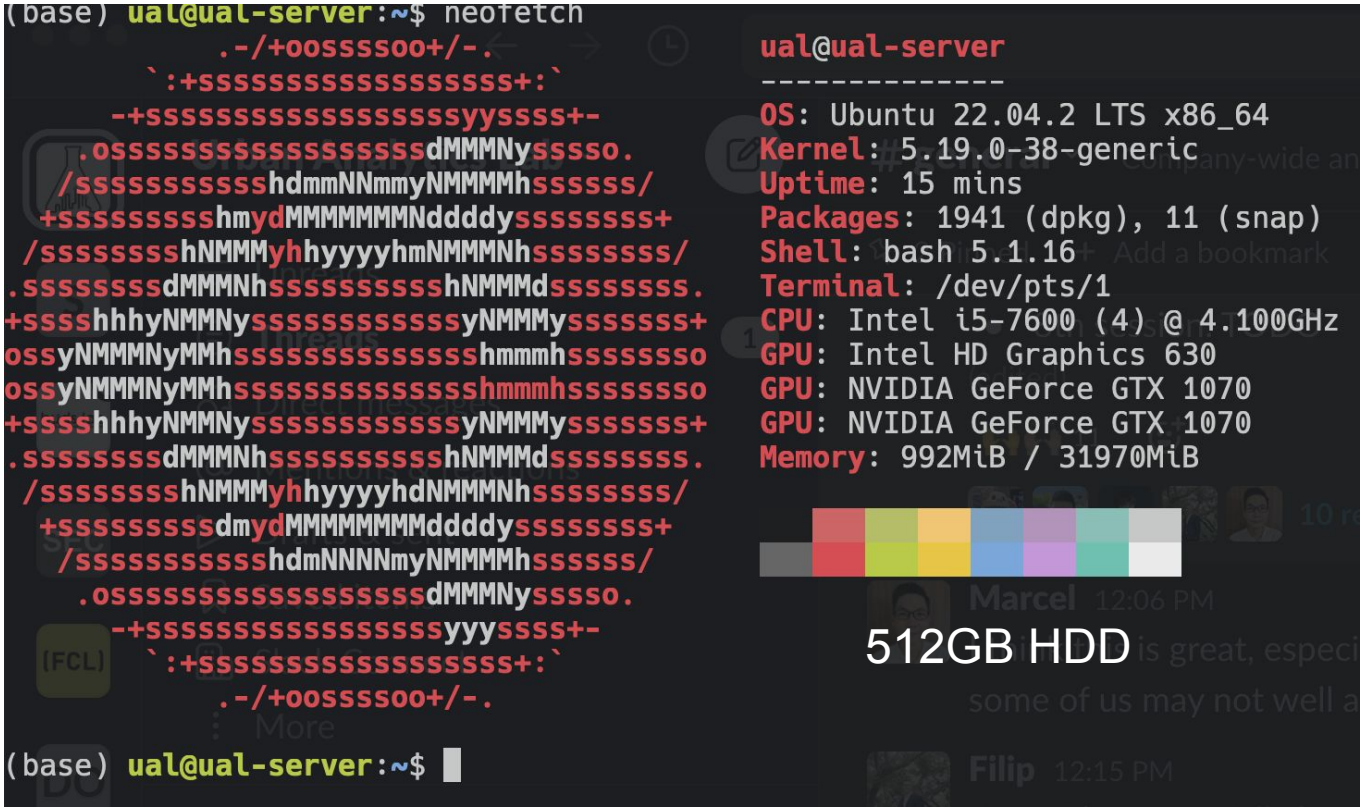
Password: **ual2023<number_of_characters_in_your_first_name>**



PLEASE change your password using **passwd**

What can I do?

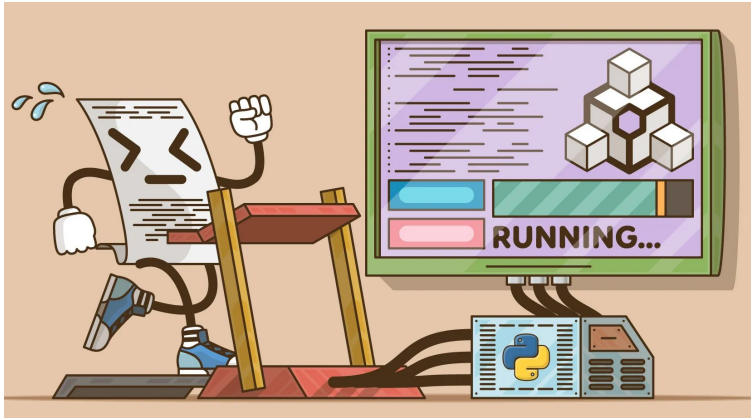
What can I do?


(base) ual@ual-server:~$ neofetch
ual@ual-server
OS: Ubuntu 22.04.2 LTS x86_64
Kernel: 5.19.0-38-generic
Uptime: 15 mins
Packages: 1941 (dpkg), 11 (snap)
Shell: bash 5.1.16
Terminal: /dev/pts/1
CPU: Intel i5-7600 (4) @ 4.100GHz
GPU: Intel HD Graphics 630
GPU: NVIDIA GeForce GTX 1070
GPU: NVIDIA GeForce GTX 1070
Memory: 992MiB / 31970MiB
512GB HDD
(base) ual@ual-server:~$

What can I do?

Train model

Moving files



Running scripts

What can I do?

In the server clone the repository:

```
git clone https://github.com/matqr/run-remote /home/<username>/run-remote/
```

Train model

Use an **ENVIRONMENT**.
Specially in a shared servers, you don't want to mess with someone else's dependencies.

Train model

Inside the cloned repo run:

```
conda env create -f environment_sandbox.yml

conda activate sandbox

python torch_example.py
```

```
(sandbox) matias@ual-server:~/run-remote$ python torch_example.py
Are we using GPUs? True
How many GPUs can you find? 2
Which device are you using? 0
What's your name? NVIDIA GeForce GTX 1070
tensor([[-2.3889, -2.3127, -2.3151, -2.1710, -2.4112, -2.2448, -2.2118, -2.3052,
         -2.4720, -2.2336]], grad_fn=<LogSoftmaxBackward0>)
(sandbox) matias@ual-server:~/run-remote$
```

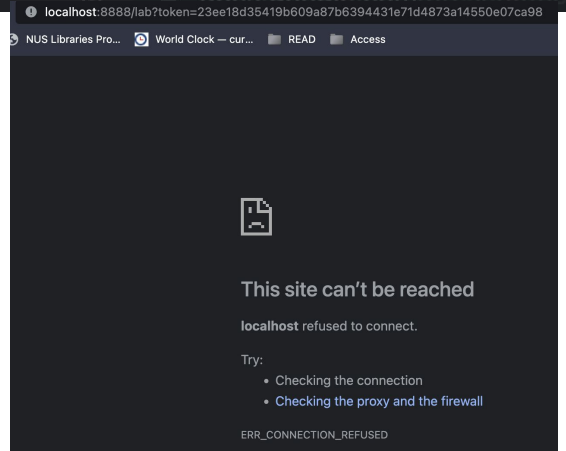
Train model

Use an **ENVIRONMENT**.
Specially in a shared servers, you don't want to mess with someone else's dependencies.

Inside the cloned repo run: **jupyter lab**

```
Refresh (1 sec) http://localhost:8888/lab?token=23ee18d35419b609a87b6394431e71d4873a14550e07ca98

This page should redirect you to a Jupyter application. If it doesn't, click here to go to Jupyter.
```

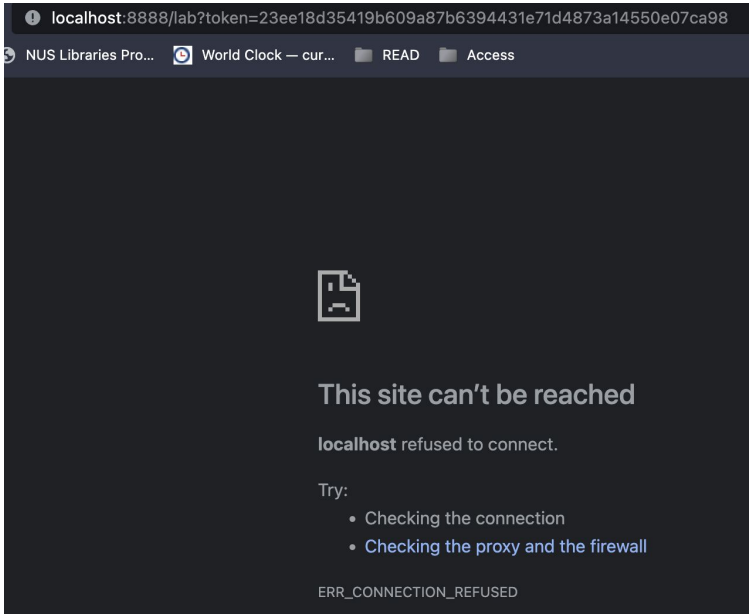


Can't see the notebook!!

Train model

Use an **ENVIRONMENT**.
Specially in a shared servers, you don't want to mess with someone else's dependencies.

`Inside the cloned repo run:` **`jupyter lab --no-browser --port=8080`**



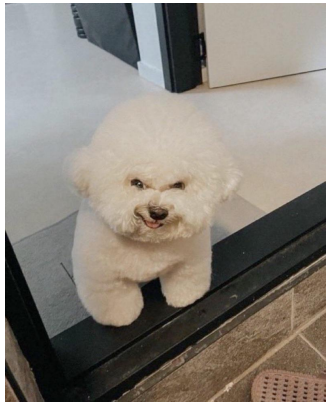
STILL?!

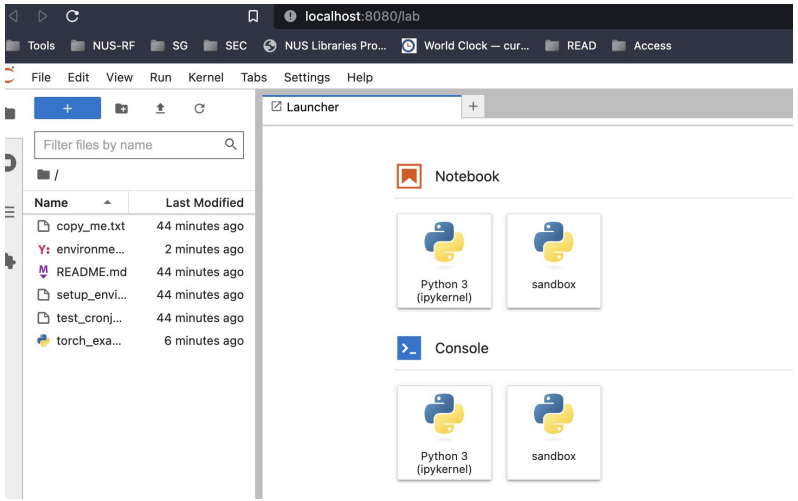
Train model

Use an **ENVIRONMENT**.
Specially in a shared servers, you don't want to mess with someone else's dependencies.

From your terminal: **ssh -L 8080:localhost:<PORT> <REMOTE_USER>@<REMOTE_HOST>**

ssh -L 8080:localhost:8080 matias@10.248.3.161





If you want to use jupyter like this, ssh to the server like this from now on.

You can use any 80xx port

Train model

Use an **ENVIRONMENT**.
Specially in a shared servers, you don't want to mess with someone else's dependencies.

But I want to keep using the terminal in the server! Should I ssh again? **NO**

Train model

Use an **ENVIRONMENT**.
Specially in a shared servers, you don't want to mess with someone else's dependencies.

Inside the cloned repo run:

```
screen jupyter lab --no-browser --port=8080
```

Train model

Use an **ENVIRONMENT**.
Specially in a shared servers, you don't want to mess with someone else's dependencies.

Detach from the screen, press:

ctrl+a+d

```
(sandbox) matias@ual-server:~/run-remote$ screen jupyter lab --no-browser --port=8080
[detached from 20501.pts-1.ual-server]
(sandbox) matias@ual-server:~/run-remote$
```

Train model

Use an **ENVIRONMENT**.
Specially in a shared servers, you don't want to mess with someone else's dependencies.

List current screens:

```
screen -ls
```

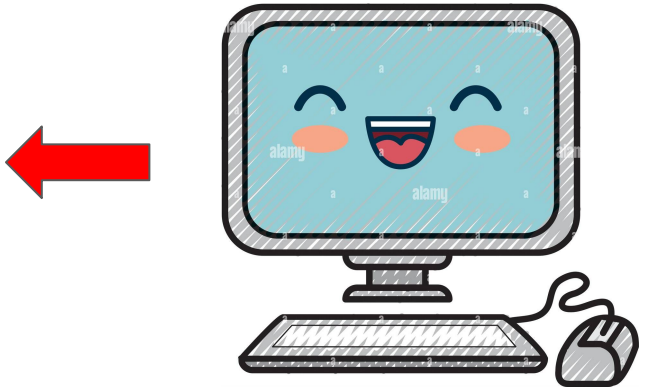
```
(sandbox) matias@ual-server:~$ screen -ls
There is a screen on:
        20501.pts-1.ual-server  (17/04/23 08:07:26)     (Detached)
1 Socket in /run/screen/S-matias.
(sandbox) matias@ual-server:~$
```

Attach to screen:

```
screen -r <screen-id>
```

Moving files

Moving files



From your computer:

scp <username>@10.248.3.161:~/run-remote/copy_me.txt .

```
 ~/Developer  scp matias@10.248.3.161:~/run-remote/copy_me.txt .                (base)
matias@10.248.3.161's password:
copy_me.txt                                   100%  271    35.3KB/s   00:00
 ~/Developer                                                                   (base)
```

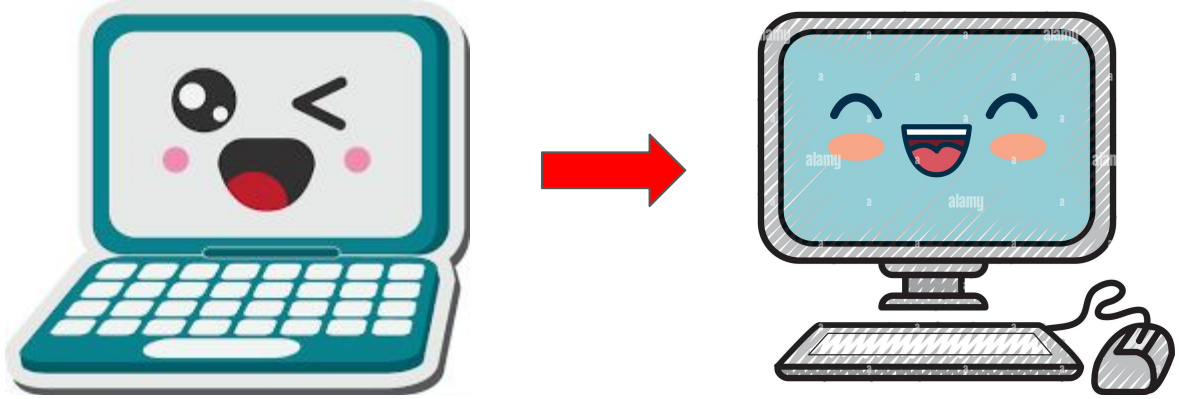
Moving files

Let’s rename the file we just copied from the server

Inside the cloned repo run: **mv copy_me.txt copied_you.txt**

```
~/Developer  mv copy_me.txt copied_you.txt                    (base)
~/Developer                                                   (base)
```

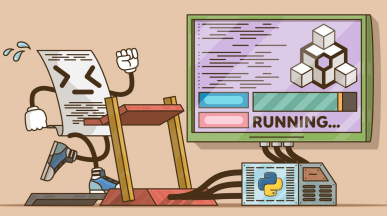
Moving files



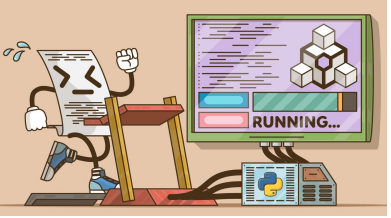
From your computer:

scp copied_you.txt <username>@10.248.3.161:~/run-remote/

```
✘ ~/Developer scp copied_you.txt matias@10.248.3.161:~/run-remote/          (base)
matias@10.248.3.161's password:
copied_you.txt                                  100%  271    39.9KB/s   00:00
 ~/Developer                                                               (base)
```



Running scripts



Running scripts

cron: job scheduler (scripts that run at specifics timestamps)

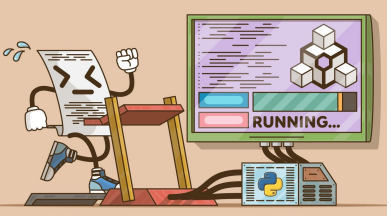
Where are they? **crontab** (text file)

Run command: **crontab -e**

```
(sandbox) matias@ual-server:~/run-remote$ crontab -e
no crontab for matias - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/nvim
  3. /usr/bin/vim.tiny
  4. /bin/ed

Choose 1-4 [1]:
```

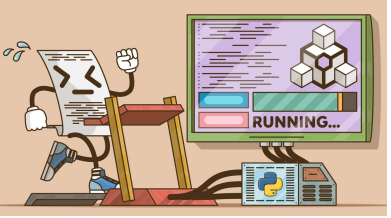


Running scripts

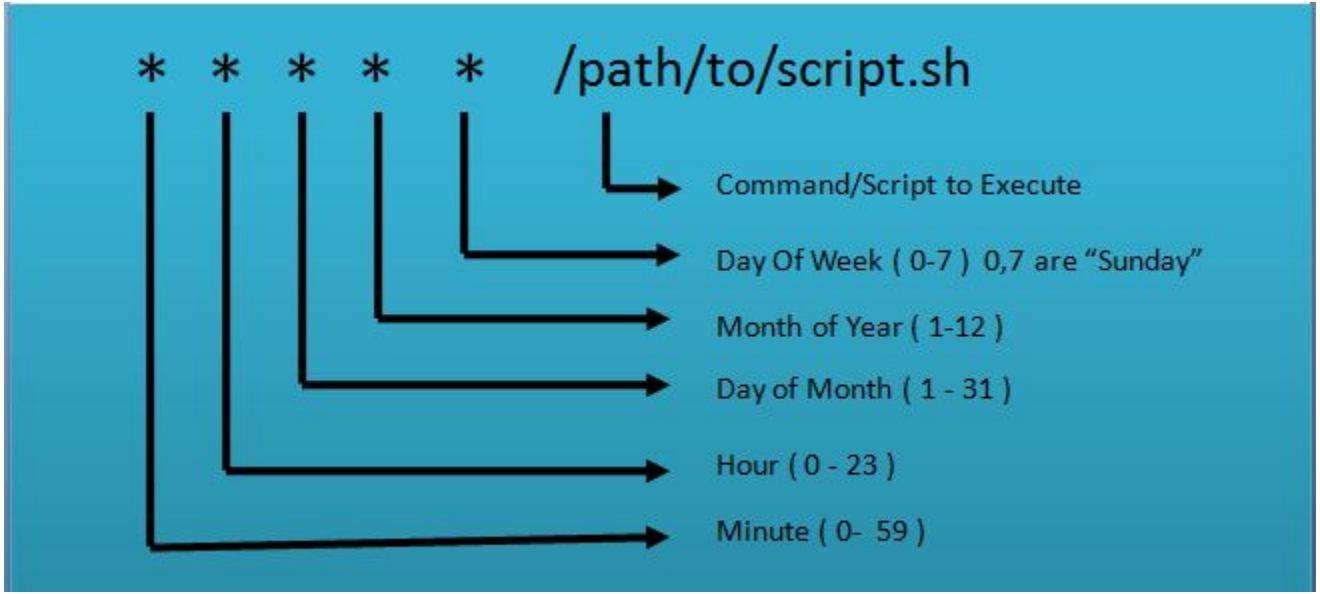
```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
```

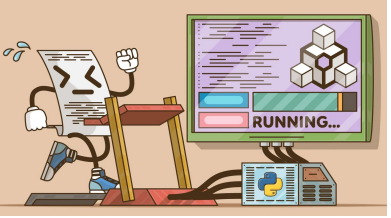
Running scripts

```
crontab.PrAl1H/crontab                                    23,1          All
```



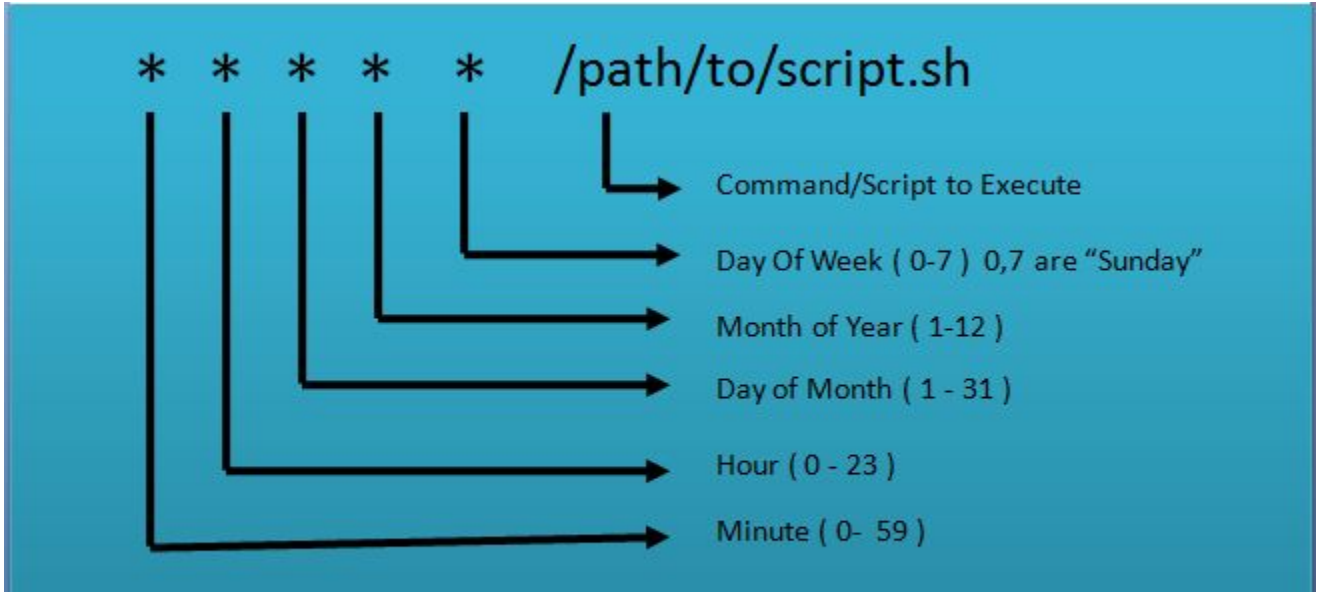
Running scripts

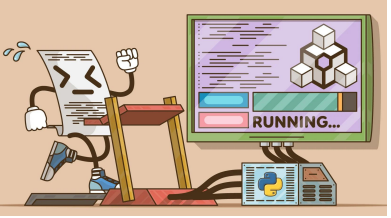




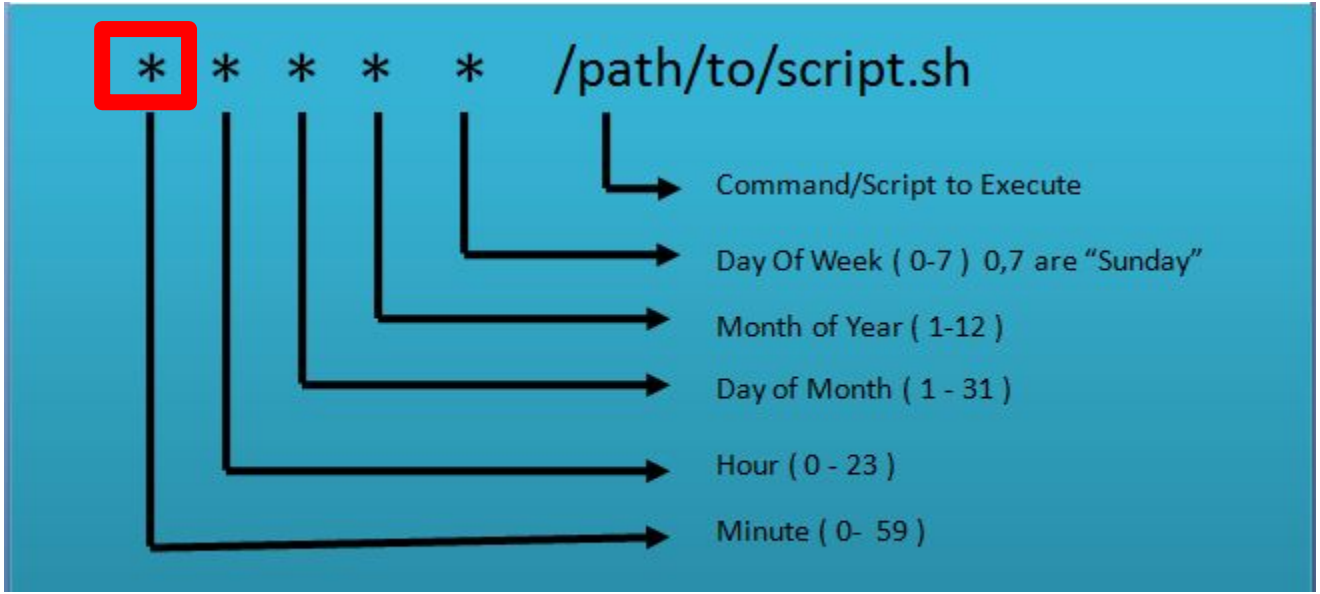
“I want to query an API every 2 minutes to give me the outside temperature on Campus (nearest weather station)”

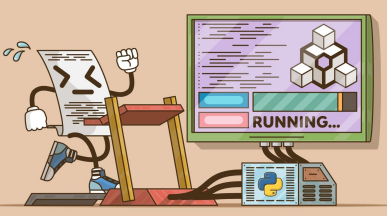
Running scripts





Running scripts

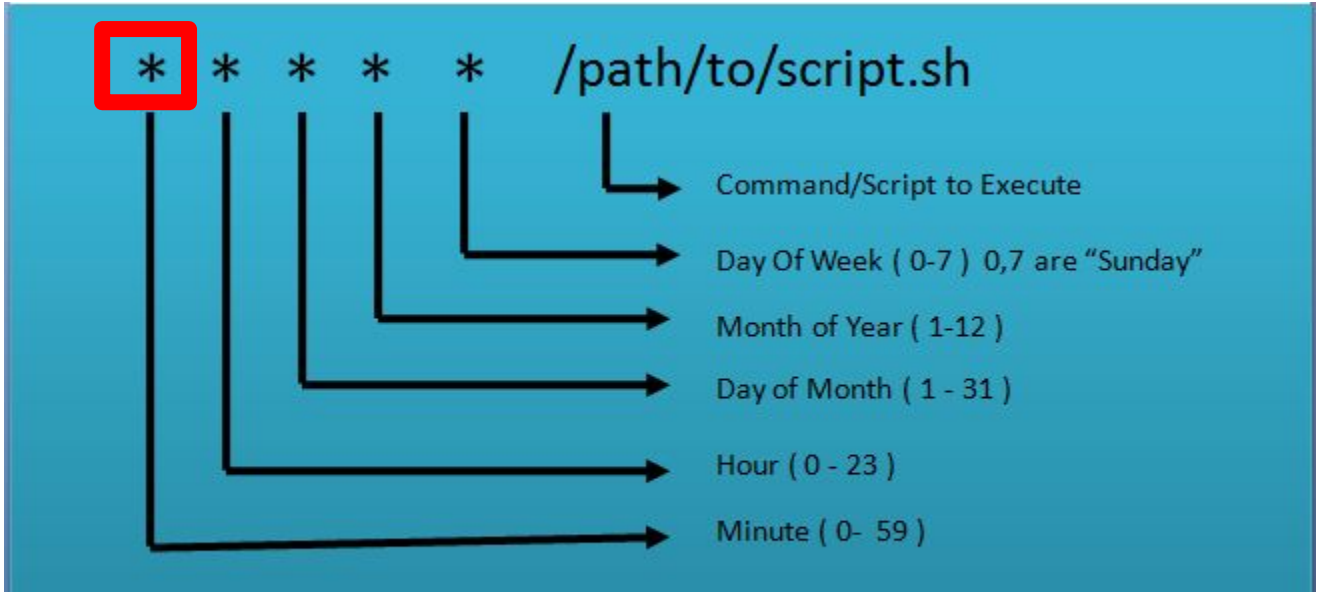


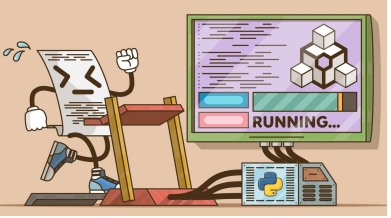


"I want to print 'first UAL workshop' to a new file, or if exist append to it, every 2 minutes"

Running scripts

```
#!/bin/sh
echo "first UAL workshop!" >> output_cronjob.txt
```





Running scripts

crontab guru

The quick and simple editor for cron schedule expressions by Cronitor

"At every minute."

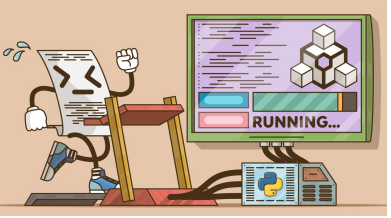
next at 2023-04-15 16:49:00 random

* * * * *

minute	hour	day (month)	month	day (week)

*	any value
,	value list separator
-	range of values
/	step values
@yearly	(non-standard)
@annually	(non-standard)
@monthly	(non-standard)
@weekly	(non-standard)
@daily	(non-standard)
@hourly	(non-standard)
@reboot	(non-standard)

https://crontab.guru/



Running scripts

crontab guru

The quick and simple editor for cron schedule expressions by Cronitor

"At every 2nd minute."

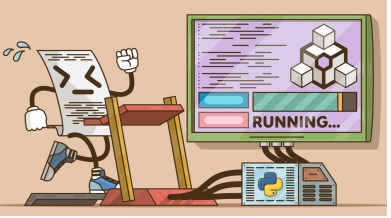
next at 2023-04-15 16:50:00

random

```
*/2  *  *  *  *
```

minute	hour	day (month)	month	day (week)
		*	any value	
		,	value list separator	
		-	range of values	
		/	step values	
		@yearly	(non-standard)	
		@annually	(non-standard)	
		@monthly	(non-standard)	
		@weekly	(non-standard)	
		@daily	(non-standard)	
		@hourly	(non-standard)	
		@reboot	(non-standard)	

https://crontab.guru/



Add to the last line:

***/2 * * * * ~/run-remote/test_cronjob.sh**

Running scripts

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/2 * * * * ~/run-remote/test_cronjob.sh
~
~
```

(More) Housekeeping

- Where can I keep learning?
  - The missing semester (https://missing.csail.mit.edu/)
  - Vim/NeoVim (https://youtu.be/w7i4amO_zaE)
  - Stackoverflow
  - ChatGPT
  - Etc

- Next Workshops
  - Use Minimum Working Examples (MWE) (i.e., github repos)
  - Keep it hands-on!