

Lab setup

Eenmalige setup

Het instellen van je labomgeving

Iedereen heeft een eigen Virtual Machine (VM) waar we op gaan werken. Iedereen heeft standaard 150 uur extra beschikbaar, dat kan worden uitgebreid als dat nodig is. Zet de VM uit als je hem niet gebruikt, het kost 0.55€ per uur per persoon dus dat tikt op een gegeven moment wel aan. Elke avond om 21:00 sluit hij bovendien automatisch af.

Deze instructies leren je:

- Een ssh-keypair aan te maken
- Via VS Code een ssh-verbinding te maken met de VM
- De code van de repository te clonen naar je server
- Een environment te maken via poetry

Installeer VS code

We gaan met VS code werken.

Dit kun je installeren via <https://code.visualstudio.com>

Registrieren

- A. Zorg dat je een microsoft account hebt, en dat je de uitnodiging voor het lab op je microsoft account ontvangt.
 - 1. Als je nog geen microsoft account hebt, kun je er een aanmaken met het e-mailadres waarop je gemaild bent
 - 2. Als je een al een ander microsoft account hebt dat je graag wilt gebruiken, geef dat e-mailadres dan door, dan stuur ik een nieuwe uitnodiging.
- B. Open de email van azure-noreply@microsoft.com met de titel "Registreer u voor het lab " en klik op de link.
- C. Als je de melding krijgt dat je geen toegang hebt: check dat je ingelogde microsoft account en het e-mailadres waarop je bent uitgenodigd dezelfde zijn.

Lokaal - Vind je terminal

- Voor Mac: zoek naar “terminal” (bv met spotlight) en open hem
- Voor Windows:
 - Installeer git bash <https://gitforwindows.org/>
 - Check dat je OpenSSH is geïnstalleerd via start > settings > apps > Optional features > zoek OpenSSH. Installeer indien nodig
 - Zoek en open “git bash”
 - Als ik het in dit document over de terminal heb, dan bedoel ik daarmee voor windows de “git bash”

VM - Start je VM

- Ga naar <https://labs.azure.com/virtualmachines>
- Klik op de button waar hier rechts “gestopt” bij staat
- Je VM wordt opgestart, dit duurt ongeveer twee minuten
- Als je “Running” of “Actief” ziet staan, klik op het kleine computer icon rechts onderin en kies “Connect via SSH”
- De eerste keer zul je een wachtwoord moeten aanmaken. Azure stelt bepaalde eisen aan je wachtwoord (hoofdletters, cijfers, tekens).
- Elke keer dat je dingen wilt aanpassen op je VM, zul je dit wachtwoord moeten invoeren. Sla het op!



VM - Log in via ssh

Als je op “Connect via SSH” op <https://labs.azure.com/virtualmachines> klikt zie je een lang commando. Plak dat in je lokale terminal.

De eerste keer krijg je de vraag of je de key-fingerprint van de server wilt toevoegen. Geef yes als antwoord.

Het wachtwoord waar om wordt gevraagd is je VM wachtwoord. Je bent nu ingelogd, zie screenshot. Dit is de terminal van de VM.

Als het goed is zie je kleurtjes. Typ anders zsh

```
ED25519 key fingerprint is SHA256:I9ZfLL0Sf7Zn5ogNAs0XpUe7Jahsk8k6bpCRy2gAfiA.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '[ml-lab-499b0cc0-f0fa-444d-a905-b2d2bead7477.westeurope.cloud  
app.azure.com]:57696' (ED25519) to the list of known hosts.  
mladmin@ml-lab-499b0cc0-f0fa-444d-a905-b2d2bead7477.westeurope.cloudapp.azure.com's passw  
ord:  
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-1022-azure x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Fri Apr 22 09:28:36 UTC 2022  
  
System load:  0.07               Processes:            144  
Usage of /:   11.1% of 28.90GB   Users logged in:     1  
Memory usage: 1%                IPv4 address for eth0: 10.0.0.12  
Swap usage:   0%  
  
0 updates can be applied immediately.  
  
Last login: Fri Apr 22 09:26:14 2022 from 62.166.163.14  
λ ML-RefVm-812132 ~ +
```

Lokaal - Maak een ssh-keypair

- Als je al een ssh-key hebt, mag je dit overslaan!
- Geef in je terminal het commando:

`ssh-keygen -C "naam@email.com"`

Waarbij je je eigen email invult (zie afbeelding)

- Je terminal vertelt je waar je key wordt opgeslagen. Start daar in

`~/.ssh/id_rsa`

wat prima is.

- Je mag je wachtwoord leeg laten, dat is wel zo handig. Mocht je een ssh-key hebben met een wachtwoord, dan kun je gewoon een nieuwe aanmaken zonder wachtwoord. Verwijder dan eerst de oude (tenzij je hem ergens voor nodig hebt, dan maak je een nieuwe met een andere naam).

```
rgrouls@NED-N-1077 MINGW64 ~  
$ ssh-keygen -C "rgrouls@outlook.com"  
Generating public/private rsa key pair.  
Enter file in which to save the key (/c/Users/rgrouls/.ssh/id_rsa):  
Created directory '/c/Users/rgrouls/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /c/Users/rgrouls/.ssh/id_rsa  
Your public key has been saved in /c/Users/rgrouls/.ssh/id_rsa.pub  
The fingerprint is:
```


Lokaal - Copy de public key

- Open lokaal (op je laptop) een tweede terminal, en navigeer naar je lokale `~/.ssh` folder met behulp van `cd`
 - Lokaal staan in je `~/.ssh` folder de ssh-keys. We gaan de public key kopiëren. Geef lokaal het commando

```
cat ~/.ssh/id_rsa.pub | clip
```

Voor windows, en voor mac:

```
pbcopy < ~/.ssh/id_rsa.pub
```


- Toelichting bij het commando:
 - In nederlands lees je dit als: print de inhoud van de public key, en stuur de output daarvan door naar mijn clipboard.
 - `cat file` print de inhoud van het bestand `file`, we printen dus de inhoud van de public key met `cat ~/.ssh/id_rsa.pub`
 - door daar `| clip` achter te zetten sturen we de output van `cat` door naar `clip`, wat je clipboard is op windows. Nu kun je de inhoud van de file ergens anders plakken
 - Voor mac laat je pbcopy (je clipboard) de inhoud ontvangen via `<` van de `~/.ssh/id_rsa.pub` file

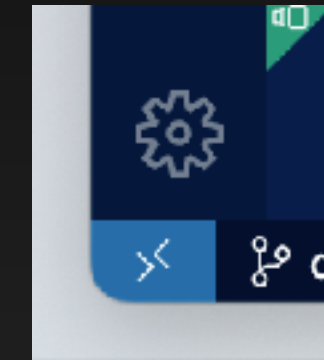
VM - Paste de public key

je kunt nu de inhoud van je public key gaan plakken naar de VM

- ga naar de terminal waarmee je op je VM bent ingelogd en navigeer naar de `.ssh` directory met het commando `cd ~/.ssh`
- In die folder staat een bestand, namelijk `authorized_keys`. Check met `ls`
- Open nu het `authorized_keys` bestand met een text editor, bijvoorbeeld nano:
`nano authorized_keys`
- Plak nu met `cmd+v` of `shift+insert` of `win+v` of je muis de inhoud van `id_rsa.pub` in het `authorized_keys` bestand op de VM

VS Code - de VM toevoegen

- Ga naar VS Code en installeer Remote-SSH extension: ga naar extensions  zoek naar *remote ssh* en klik *install*.
- Linksonder in je VS code kun je nu verbinding maken via de blauwe pijltjes
- Kies “connect to Host... Remote-SSH”
- Kies “+ Add New SSH Host...”
- Copy-paste het commando om met ssh te verbinden dat je op <https://labs.azure.com/virtualmachines> via “Connect via SSH” krijgt (zie slide “VM - login via SSH”)



Enter SSH Connection Command

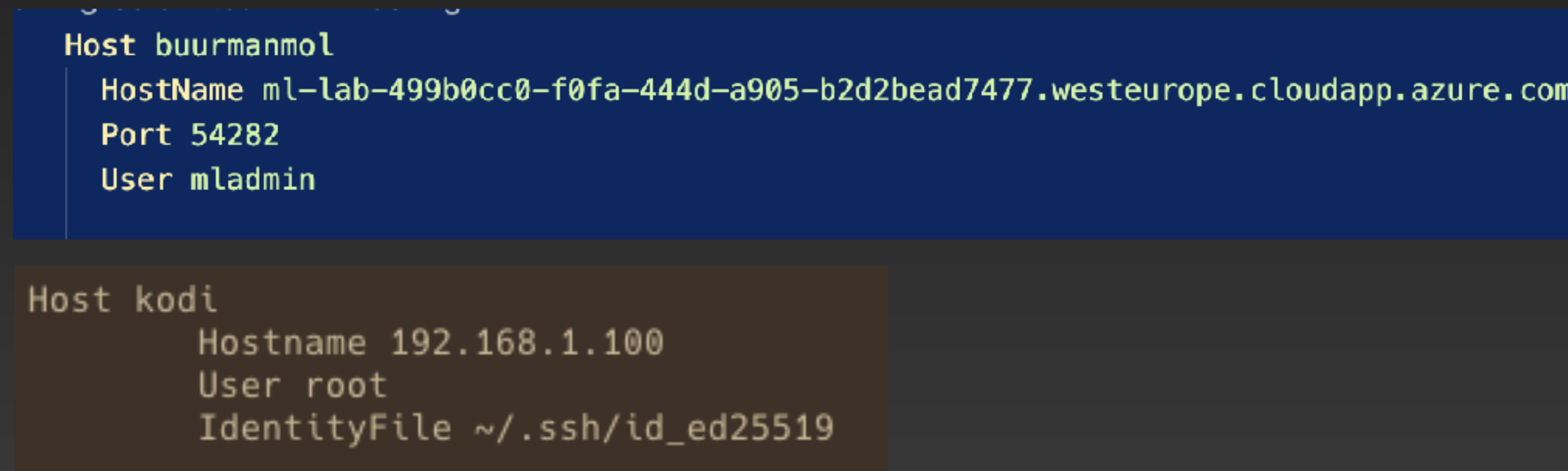
ssh hello@microsoft.com -A

Press 'Enter' to confirm your input or 'Escape' to cancel

- Bij “Select SSH configuration file to update”, kies je het .ssh/config bestand in je eigen gebruikersaccount

VS code verbinden

- Je kunt nu eventueel de naam wat makkelijker maken. Dat doe je door via VS code je configuratiefile aan te passen:
- Zoek door de commandos met `ctrl+shift+P` of `cmd+shift+P`
- zoek "Remote-SSH Open SSH Configuration File..."
- Pas de naam na Host aan naar een makkelijk te onthouden naam
- Als jouw key een andere naam heeft dan standaard, dan kun je dat in je config toevoegen via `IdentityFile`



Ok, dat was een heleboel!

Als je tot hier gekomen bent, en alles gelukt is, dan is de basis klaar!

De voorgaande paginas hoef je in principe niet meer te gebruiken.



In principe waren de meeste van deze stappen eenmalig, en is het inloggen voortaan een stuk eenvoudiger...

Workflow

Inloggen op je server is voortaan relatief simpel:

- Start je machine op via <https://labs.azure.com/virtualmachines>
- Verbind VS code via “Remote-SSH: Connect to Host...”
- Kies de naam van de server
- geef aan dat de VM een linux server is als daarom wordt gevraagd.
- Open een folder om in te werken.
- Als je klaar bent, sluit de verbinding via “Close remote connection”
- Stop je machine weer via de website

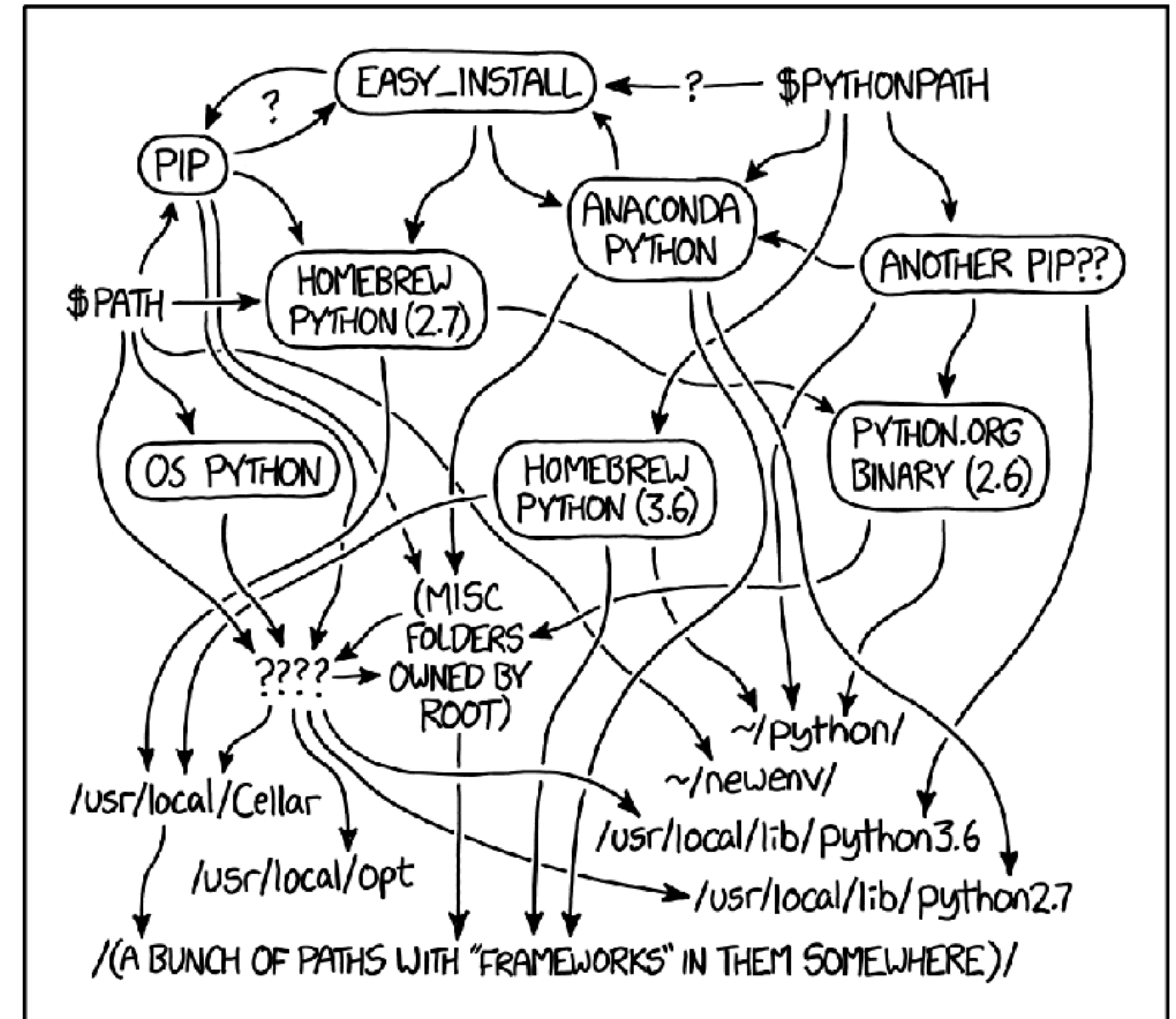
VM - Installeer de omgeving

Veel van jullie zijn waarschijnlijk gewend om met conda of pip te werken.

Dat gaan wij niet doen:

We gebruiken rye: <https://rye.astral.sh/>

Dit managed zowel je python versies eenvoudig, als je virtual environments



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

VM - setup

- In de root directory van de machine vind je de cursusfolder. Open VScode in die folder. Dit is belangrijk! Als je je project in de parent folder opent werkt VScode niet goed met je python omgeving samen.
- In de projectfolder staat een `pyproject.toml` bestand. Dit beschrijft de dependencies die we gebruiken. Het is leesbaar, dus bekijk het gerust met `vscode` (of gebruik `cat`)
- Run ``git pull`` in de projectfolder. Het is mogelijk dat ik nog wat laatste wijzigingen heb gedaan.

VM - setup

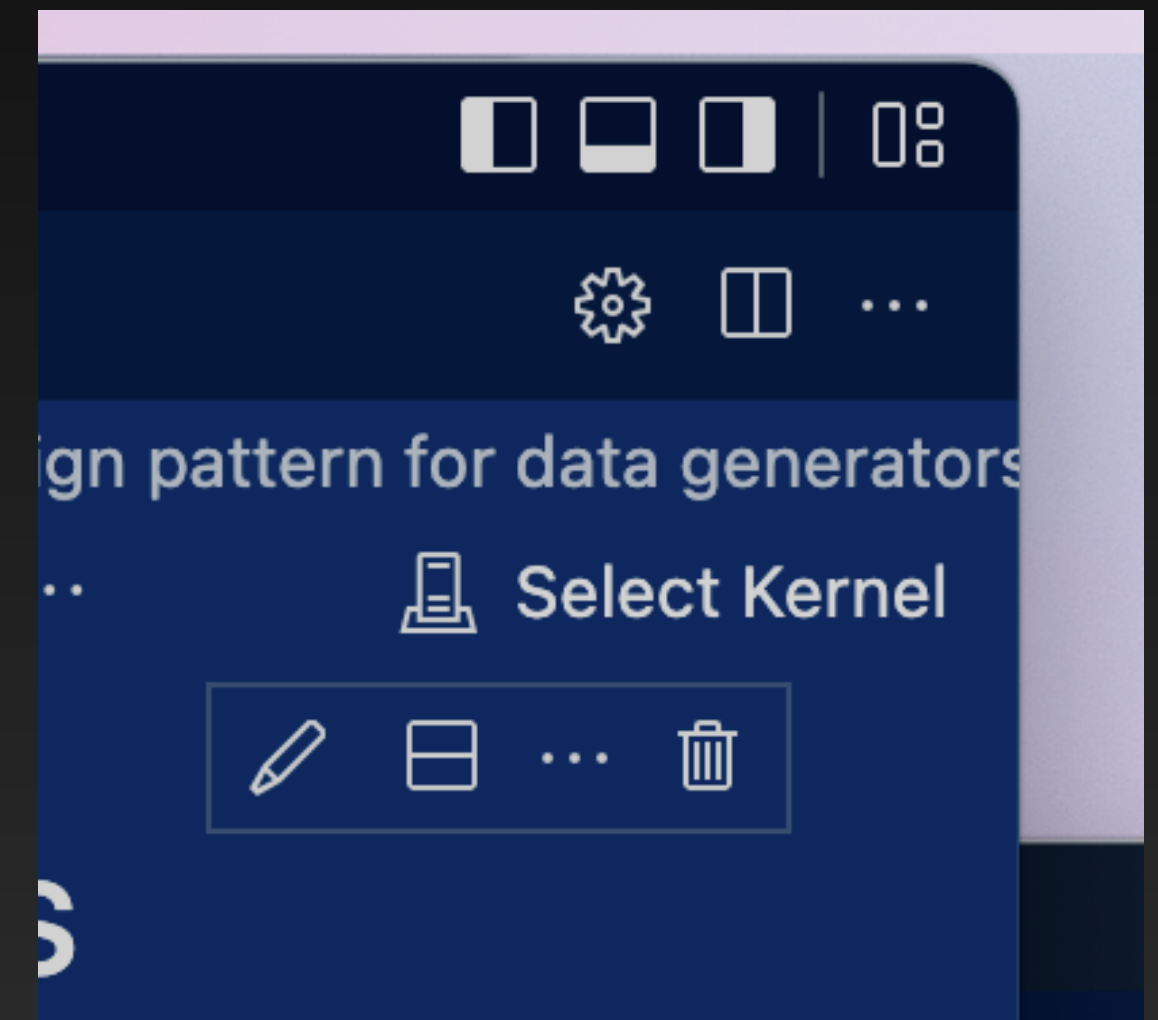
- Check of rye is geïnstalleerd met `which rye`, wat een locatie moet terugsturen, of `rye --help`, wat uitleg moet geven
- Als dat niet het geval is, volg de instructies op <https://rye.astral.sh/> om rye te installeren:
 - Run `curl -sSf https://rye.astral.sh/get | bash`
 - Voeg rye toe aan PATH als de installer daarom vraagt
 - Als de installer vraagt om een standaard python, zeg `3.11`
 - Run de eerste keer na installatie `source "$HOME/.rye/env"`

VM - Rye

- Kijk de instructie video van rye <https://rye.astral.sh/guide/>
- In plaats van ``pip install xyz`` doe je voortaan ``rye add xyz`` waarbij `xyz` de naam van de dependency is, bijvoorbeeld ``pdm add numpy``.
- Alles wat je toevoegt wordt aan je `pyproject.toml` file toegevoegd, automatisch.
- Er is voor dit project al een `pyproject.toml` file, en je wilt geen dependencies aan mijn lesmateriaal toevoegen (doe dat ik je eigen project). Wat je wel wilt doen is een `.venv` aanmaken met `rye sync`
- Als je je environment wilt gebruiken vanuit de shell, activeer dan je environment met `source .venv/bin/activate`

In jupyter een kernel selecteren

- Open een jupyter notebook.
- Je kunt nu de kernel selecteren in je jupyter notebooks: klik rechtsboven op “**select kernel**”, eventueel **`Python environments`** eerst.
- Dit werkt ALLEEN goed als je 1) een **.venv** hebt aangemaakt 2) VScode in de juiste folder hebt geopend.
- Waarschijnlijk herkent vscode je **`venv`** folder automatisch en beveelt deze aan met een ster *
- Zo niet, herstart VScode (ctrl+shift+p reload window)
- Je kunt nu notebook runnen vanuit de betreffende environment



Werken met git

- Git is in het begin verwarrend, als je daar nog nooit mee hebt gewerkt. Maar het is nu eenmaal (bijna) de enige reële manier als je met meerdere mensen tegelijkertijd aan code wilt schrijven.
- Jullie kunnen geen code pushen naar mijn repository. Maar je kunt wel branchen. Als je niet weet wat push of branch betekent, lees de git crash course.
- Als je al wilt beginnen met dingen testen, doe dat in je eigen branch:
 - Open de terminal van je VM
 - navigeer naar de lesfolder
 - maak een nieuwe branch genaamd `les1` met `git checkout -b les1`
De `-b` heb je alleen de 1e keer nodig.
 - Je ziet nu dat je niet meer in master zit, maar in `les1`, je kunt nu veilig werken en nog steeds updates ontvangen van de lessen.

```
rename notebooks/{1_pytorch_intro => 2_convolutions}/06_resnet.ipynb (100%)
λ ML-RefVm-812132 ML22 → λ git master → git checkout -b les1
Switched to a new branch 'les1'
λ ML-RefVm-812132 ML22 → λ git les1 → poetry update
Updating dependencies
```


Werken met git

- Ik maak regelmatig updates aan de code. Je kunt de laatste versie altijd binnenhalen via `git pull`. Minimale instructies:
 - Sla je werk op in je les1 branch:
`git checkout les1`
`git add .`
`git commit -m "mijn eerste commit"`
 - ga terug naar master en pull:
`git checkout master`
`git pull`
 - Je bent je werk niet kwijt. Dat zit in les1. Als je daarnaar terug wilt, doe je weer
`git checkout les1`

```
rename notebooks/{1_pytorch_intro => 2_convolutions}/06_resnet.ip
λ ML-RefVm-812132 ML22 → λ git master → git checkout -b les1
Switched to a new branch 'les1'
λ ML-RefVm-812132 ML22 → λ git les1 → poetry update
Updating dependencies
```

Werken met git

- Life-Pro-Tip: geef je eigen notebooks een andere naam dan de mijne. Op die manier krijg je nooit conflicten:
 1. Copy een notebook en hernoem het
 2. Werk in je notebook, sla de wijzigingen op via git commit.
 3. Als alles is gecommit, kun je terug naar master, een git pull doen, weer terug naar je eigen branch en 'git merge master' doen. Alle updates worden dan geïntegreerd met je eigen branch.

PDM update

- Ik voeg af en toe nieuwe libraries toe. Om te zorgen dat je die ook allemaal op jouw server krijgt, moet je af en toe de boel bijwerken.
- Dat gaat eenvoudig door:
 - Te navigeren naar ~/code/MADS-cursus (waarbij je MADS-cursus vervangt door de naam van de folder)
 - `pdm update` te runnen. PDM werkt dan alle dependencies bij.

```
λ ML-RefVm-812132 ML22 → λ git les1 → poetry update
Updating dependencies
Resolving dependencies... (67.2s)

Package operations: 3 installs, 12 updates, 0 removals

• Updating pygments (2.11.2 -> 2.12.0)
• Updating google-auth (2.6.5 -> 2.6.6)
• Updating protobuf (3.20.0 -> 3.20.1)
• Updating gast (0.4.0 -> 0.5.3)
• Updating keras (2.7.0 -> 2.8.0)
• Updating libclang (13.0.0 -> 14.0.1)
```

Crash course command line

De enige commando's die je hier kunt geven zijn tekst commando's. Ik heb een paar tweaks toegevoegd om het leven makkelijker te maken.

- `ls` : List. Geeft een overzicht van alle zichtbare bestanden. Blauw zijn directories. Probeer ook `lsd`
- `la` : shortcut voor `ls -A` , toont ook verborgen bestanden (die beginnen met een `.`)
- `cd` : Change directory
- `pwd` : Present Working Directory
- `mkdir` : Make directory.
- `df -H` : laat gebruikte schijfruimte zien
- `du -sh *` : laat de schijfruimte voor deze folder zien

```
λ ML-RefVm-812132 ~ → la
total 176K
-rw----- 1 mldadmin mldadmin 429 Apr 22 09:26 .bash_history
-rw-r--r-- 1 mldadmin mldadmin 220 Feb 25 2020 .bash_logout
-rw-rw-r-- 1 mldadmin mldadmin 67 Apr 19 14:00 .bashrc
-rw-r--r-- 1 mldadmin mldadmin 3.7K Feb 25 2020 .bashrc.bak
drwx----- 3 mldadmin mldadmin 4.0K Apr 19 14:08 .cache
drwxrwxr-x 4 mldadmin mldadmin 4.0K Apr 19 14:08 .local
drwxr-xr-x 12 mldadmin mldadmin 4.0K Apr 19 13:50 .oh-my-zsh
-rw-r--r-- 1 mldadmin mldadmin 904 Apr 19 13:50 .profile
drwxrwxr-x 14 mldadmin mldadmin 4.0K Apr 19 14:02 .pyenv
drwx----- 2 mldadmin mldadmin 4.0K Apr 22 09:32 .ssh
-rw-r--r-- 1 mldadmin mldadmin 0 Apr 19 13:48 .sudo_as_admin_successful
-rw----- 1 mldadmin mldadmin 2.9K Apr 22 09:32 .viminfo
-rw-rw-r-- 1 mldadmin mldadmin 48K Apr 19 13:54 .zcompdump
-rw-rw-r-- 1 mldadmin mldadmin 50K Apr 19 14:09 .zcompdump-ML-RefVm-812132-5.8
-rw-rw-r-- 1 mldadmin mldadmin 97 Apr 19 13:50 .zprofile
-rw----- 1 mldadmin mldadmin 1.9K Apr 22 09:36 .zsh_history
-rw-rw-r-- 1 mldadmin mldadmin 4.1K Apr 19 14:09 .zshrc
-rw-rw-r-- 1 mldadmin mldadmin 4.1K Apr 19 14:09 .zshrc.bck
-rw-rw-r-- 1 mldadmin mldadmin 23 Apr 19 13:50 .zshrc.pre-oh-my-zsh
drwxrwxr-x 3 mldadmin mldadmin 4.0K Apr 19 13:48 serverinstall
λ ML-RefVm-812132 ~ →
```