

ubuntu

secure shell

ssh user@(x.x.x.x Ip Address)

คำสั่ง

* ctrl z = ย้อนกลับ

* clear = เคลียร์หน้าจอ

* list = show ชื่อไฟล์ที่อยู่ใน folder

> ls-l = show file แบบยาว ของ folder

* r read อ่านได้

* x execute รันได้

* w write เขียน/แก้ไขไฟล์ได้

* ls -lt = เรียงไฟล์ตามเวลา ใหม่อยู่บน

* ls -lth = อ่านได้ง่ายขึ้น (human read)

* ls -ltr อ่านธรรมชาติ

* cd = เปลี่ยน directory

> จะดึงกลับมาหน้า home User's folder

* mkdir = สร้างโฟลเดอร์

* cd .. = ย้อนกลับมา 1 folder

* pwd เพื่อดู path ที่เราอยู่

* cd ../ = กระโดดถอยหลังกลับไปอีก folder หนึ่งได้

* cd ../../ = ถอยกลับมา 3 โฟลเดอร์

* vi สร้างไฟล์ Editor ไม่จำเป็นต้องเลือก folder

> มีสองโหมด modify & command

* i -> insert แก้ไขได้

ใส่ colon ด้านล่าง แล้ว command

* :wq = write quit ออกมาจากไฟล์

* :q! ออกโดยไม่เซฟ

* cp ./ ชื่อไฟล์ = copy file ต้นทาง / ปลายทาง

. ที่นี้

.. ย้อนไป 1

* mv คือย้ายไฟล์ / ตั้งแต่ต้นจะทำให้เอาทั้ง path

* mv ./abc.txt ./test_lv2/test2_lv3/.

* mv ชื่อไฟล์ที่ต้องการย้าย

* rm ลบไฟล์ rm ./test_lv2/test2_lv3/abc.txt

* chmod 400 ตามด้วยชื่อไฟล์

> 4 เปลี่ยนเป็น เลขเวล 4 00 ไม่ให้อ่านเลย

คาบเรียนที่ 2

* พิมพ์ python จะขึ้น anaconda linux

* สามารถใช้คำสั่ง python ได้ปกติ

* exit() เป็น exit ออกมาจาก python

* run ด้วย python ชื่อไฟล์ จะแสดงโปรแกรมออกมา

* jupyter notebook ไม่ได้ใช้ของ google ในการรันแล้ว cloud ใหม่

* tunnel คือ คอมไปเลือกมาจากอีกเครื่องนึง ssh -L คือ สร้างลิงค์มาไว้ จะสร้าง session มา 1 session คือเหมือนเปิดคอมพิวเตอร์ขึ้นมา

* ถ้า lost connection จะทำให้โดนตัดเลย ssh -> vm ช่วยสร้าง session ที่รันใน background ได้ จะสามารถรันเรื่อย ๆ ได้

* ssh -L clientport:localhost:forward

* คำสั่ง Screen

* screen -S [ชื่อ Screen] : การสร้าง session ใหม่

* screen -R [ชื่อ Screen] : การเปิด screen ใหม่ / reconnect session

* ctrl a+d เป็นการย้อนกลับ screen

* ctrl+ a k y เป็นการเอา screen นี้ออกไป

* screen -R กด tab 2 ครั้ง จะบอกว่ามี screen อะไรบ้าง

*type jupyter notebook

* ออกมาจาก refresh -> ctrl+c

* ssh -L 8866 การลิงค์ ssh -L 8866:localhost:8888 -i /home/supidchaya/vm/vm.pem supi@20.24.217.203

* เปลี่ยนชื่อเว็บ 8888 เป็น 8866

สร้าง virtual environment เพื่อให้สามารถ run ได้ โดย install package ได้ไม่กระทบกัน

* url : <https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>

To create an environment:

* conda create --name [ชื่อที่ตั้ง] python=3.x

* -- ชื่อเต็ม

* conda activate ai_prototype_env เพื่อเปิดการใช้งาน

* conda deactivate เพื่อปิดการใช้งาน

library -> install packages

* pip install [library] / conda install [library]

* python -> import pandas เพื่อเช็คว่ามีมัย

* privilege -> ความเป็นส่วนตัว

* สามารถฝึกเวอร์ชันได้เลย pip install keras==2.3.1 ถ้าอยู่ใน env

* conda env list ใช้เพื่อดูว่า มี env อะไรบ้าง

* conda list เพื่อดูว่ามี packages ตรงไปเปล่า

* conda env remove --name ENVIRONMENT ใช้ลบ env

linux create user

* url: <https://linuxize.com/post/how-to-create-users-in-linux-using-the-useradd-command/>

* htop จะบอก user ว่าใครใช้งานอยู่บ้าง

* ssh [ชื่อ userเพื่อน]@[ipเครื่องตัวเอง]

* /\$ จุดเริ่มต้นทั้งหมด จุดเริ่มต้นของไดรฟ์

* sudo -> super user do ต้องใช้สิทธิ์ของ super user เท่านั้น!!!

* github อาจารย์: <https://github.com/tohnperfect/Alprototype2022>

* add Collaborators

* git clone ก็อปโค้ดทั้งหมด

* git status เกิดอะไรขึ้นบ้าง ถ้าเกิดสีแดงแปลว่า ต้อง commit ถ้าจะเซฟลง github

* ขั้นตอนการ commit

* git add [ชื่อไฟล์]

* git commit -m "คำที่ต้องการ comment"

เฉพาะครั้งแรก

* git config --global user.email "you@example.com"

* git config --global user.name "Your Name"

* git push ใช้เพื่อให้เป็นเวอร์ชันปัจจุบัน และตรงกับ super user (** ควรใช้ก่อนแก้ไขได้**)

* Link ส่งงาน https://docs.google.com/spreadsheets/d/1SNteQ7DOU-mWRqCqTQKS8N1vn0O-L87LC3acLu17VRI/edit?fbclid=IwAR01qklq0uW_tucjA9hgM6JlbugDKnTPCX7k0ez9BhUQI3-Hv61gMcREWs#gid=0

* code README.md

* view -> open command ... -> open preview the slide

* pass: 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