○ 시스템 사양
CPU : i5-8500
VGA : GTX 1660 6G
DRAM : 16G
SSD 240G

○ ubuntu 설치

 ubuntu 16.04 LTS AMD64 버전
 컴퓨터이름 : 2team, 사용자명 : team, 사용자암호 : 1234

○ 드라이버 설치

 네트워크(와이파이 공유기)
1) iptime A3000UA
   - 드라이버 다운로드 :
https://github.com/cilynx/rtl88x2BU_WiFi_linux_v5.3.1_27678.20180430_COEX20180427-5959
2) dkms deb 패키지
   - 다운로드 :
https://ubuntu.pkgs.org/16.04/ubuntu-updates-main-amd64/dkms_2.2.0.3-2ubuntu11.8_all.deb.html
3) dkms 패키지 설치
   - $ dpkg -i dkms_2.2.0.3-2ubuntu11.8_all.deb
4) dkms 패키지를 이용한 모듈 설치
   - $ sudo dkms add ./rtl88x2BU_WiFi_linux_v5.3.1_27678.20180430_COEX20180427-5959
   - $ sudo dkms install -m rtl88x2bu -v 5.3.1
   - $ sudo modprobe 88x2bu
5) 리부팅 후 네트워크 연결 확인
   - $ sudo reboot
 기본 설정
   - $ sudo apt-get update
   - $ sudo apt-get upgrade
   - $ sudo apt-get install git-core
   - $ sudo apt-get install vim
   - $ sudo apt-get install ssh
 ndivia 드라이버(ver. 418.xx)
1) 시스템 사양 확인 => VGA 확인
   - $ sudo lspci -k
2) 우분투 버전 확인 및 변수 생성
   - $ release="ubuntu"$(lsb_release -sr | sed -e "s/\.//g")
   - $ echo $release
3) 설치 레퍼지토리 추가

- $ sudo apt install sudo gnupg
- $ sudo apt-key adv --fetch-keys "http://developer.download.nvidia.com/compute/cuda/repos/"$release"/x86_64/7fa2af80.pub"
- $ sudo sh -c 'echo "deb http://developer.download.nvidia.com/compute/cuda/repos/'$release'/x86_64 /" > /etc/apt/sources.list.d/nvidia-cuda.list'
- $ sudo sh -c 'echo "deb http://developer.download.nvidia.com/compute/machine-learning/repos/'$release'/x86_64 /" > /etc/apt/sources.list.d/nvidia-machine-learning.list'
- $ sudo apt update

4) ndivia 설치 가능 버전 확인
- https://www.nvidia.com/Download/Find.aspx?lang=kr 에서 확인가능
- 또는
- $ apt-cache search nvidia

5) 드라이버 설치
- $ sudo apt-get install nvidia-418

6) 공통 패키지 설치
- $ sudo apt-get install dkms nvidia-modprobe

7) nvidia 커널 드라이버가 사용중인지 확인
- $ sudo lspci -k
- Kernel driver in use: nvidia <- 확인

8) 드라이버 버전 확인
- $ sudo cat /proc/driver/nvidia/version
- $ nvidia-smi

9) 재부팅
- $ sudo reboot

○ 파이썬 설치(ver. 3.7.7)

1) 기본 소프트웨어 설치
- $ sudo apt-get install software-properties-common
- $ sudo add-apt-repository ppa:deadsnakes/ppa
- $ sudo apt update

2) 파이썬 버전 확인
- $ python -V
- python 2.7

- $ python3 -V
- python 3.5

3) 파이썬 설치(3.7 버전)
- $ sudo apt install python3.7

4) 파이썬 버전 선택

- $ sudo update-alternatives --config python 확인 후, 3.7 버전 선택
- 에러 발생 시 : update-alternatives: error: no alternatives for python
  - $ sudo update-alternatives --install /usr/bin/python python /usr/bin/python2.7 1
  - $ sudo update-alternatives --install /usr/bin/python python /usr/bin/python3.7 2
  - $ sudo update-alternatives --config python 확인 후, 3.7 버전 선택

5) 현재 버전 확인
- $ python --version
- $ python3 --version

6) pip 모듈 설치
- $ sudo apt-get install python3-pip
- $ pip install --upgrade pip

7) 재부팅
- $ sudo reboot

○ 쿠다설치 (ver. 10.0)

1) CUDA 10.0 버전 설치
- $ sudo apt-get install cuda-10-0
- $ sudo apt-get install libcudnn7-dev

2) 버전 확인 1
- $ cat /usr/local/cuda/version.txt

3) 재부팅
- $ sudo reboot

4) 환경경로 설정
- $ vim ~/.bashrc
  - 맨 밑줄에 추가입력
  - export PATH
  - export PATH=/usr/local/cuda/bin:$PATH
  - export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/lib64:$LD_LIBRARY_PATH
- $ source ~/.bashrc

5) 버전 확인 2
- $ nvcc -V
- or
- $ nvidia-smi

○ 주피터 설치

1) 가상환경 안에서 주피터 설치
   - $ pip3 install jupyter
2) 주피터 폴더 설정
   - $ mkdir jupyter-workspace
3) 주피터 실행 설정 파일(jupyter config) 생성
   - $ jupyter notebook --generate-config
4) 서버 비밀번호 생성
   - $ python

- >>> from notebook.auth import passwd
- >>> passwd()
- Enter password: 1234 # 암호 입력
- Verify password: 1234 # 암호 재입력
- 'sha1:a1s2d3f4...' # 입력한 비밀번호 암호화 <- 따로 저장
- >>> exit()

5) 실행시 설정 변경
   - vim ~/.jupyter/jupyter_notebook_config.py
   - 048라인 : c.NotebookApp.allow_origin = '*' # 외부 접속 허용하기
   - 204라인 : c.NotebookApp.ip = '192.168.0.14' #아이피 설정(현재 아이피)
   - 266라인 : c.NotebookApp.notebook_dir = u'/home/team/jupyter-workspace' #작업경로 설정
   - 272라인 : c.NotebookApp.open_browser = False # 시작 시 서버PC에서 주피터 노트북 창이 열릴 필요 없음
   - 281라인 : c.NotebookApp.password = u'sha1로 시작하는 암호...' #비밀번호 설정
   - 292라인 : c.NotebookApp.port = 8888 #포트 설정
6) 주피터 실행
   - $ jupyter notebook --config ~/.jupyter/jupyter_notebook_config.py

○ 도커 설치

○ 마리아DB 설치

○ 마리아DB 실행

- $ docker start mairadb
- $ docker exec -it mariadb bash
- $ mysql -u root -p1234 -> root 권한
- $ show grants for ‘podong’@’192.168.0.62’; -> podong 권한 보기
- $ show databases; -> DB 보기
- $ create user ‘podong’@’192.168.0.30’ identified by ‘1234’; -> IP 를 가진 podong 생성
- $ grant all on podong1.*to ‘podong’@’192.168.0.30’; -> podong1의 권한을 podong에

○ 가상환경 만들기

1) 가상환경 구축 라이브러리
   - pip install virtualenv
2) 가상환경 저장 경로
   - mkdir /home/team/env && cd /home/team/env
3) 가상환경 생성
   - virtualenv [name] --python=python3.7
4) 가상환경 접속
   - cd [name]/bin
   - source /.activate
5) 가상환경 나가기 & 다시 접속
   - deactivate
   - source env/[name]/bin/activate

○ 텐서플로우 설치

1) 텐서플로우-GPU 2.0.0 설치
    - pip3 install tensorflow-gpu==2.0.0
2) 확인작업
    - In[1] : import tensorflow as tf
    - In[2] : tf.test.is_gpu_available() -> True : 성공 , False : 실패
    - In[3] : with tf.device('/GPU:0'):
    - 정상 작동되면 성공

```
a = tf.constant([[1,2],[3,4]])
b = tf.constant([[5,6],[7,8]])
c = tf.matmul(a,b)
print(c)
```

○ VS Code 설치

1) curl 설치
    - sudo apt-get install curl
2) 마이크로소프트 GPG 키를 다운로드하여 /etc/apt/trusted.gpg.d/ 경로에 복사
    - sudo sh -c 'curl https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > /etc/apt/trusted.gpg.d/microsoft.gpg'
3) VS Code를 다운로드 받기 위한 저장소 추가
    - sudo sh -c 'echo "deb [arch=amd64] https://packages.microsoft.com/repos/vscode stable main" > /etc/apt/sources.list.d/vscode.list'
4) 패키지 목록 가져오기
    - sudo apt-get update
5) Visual Studio Code 설치
    - sudo apt-get install code
6) 실행
    - code