

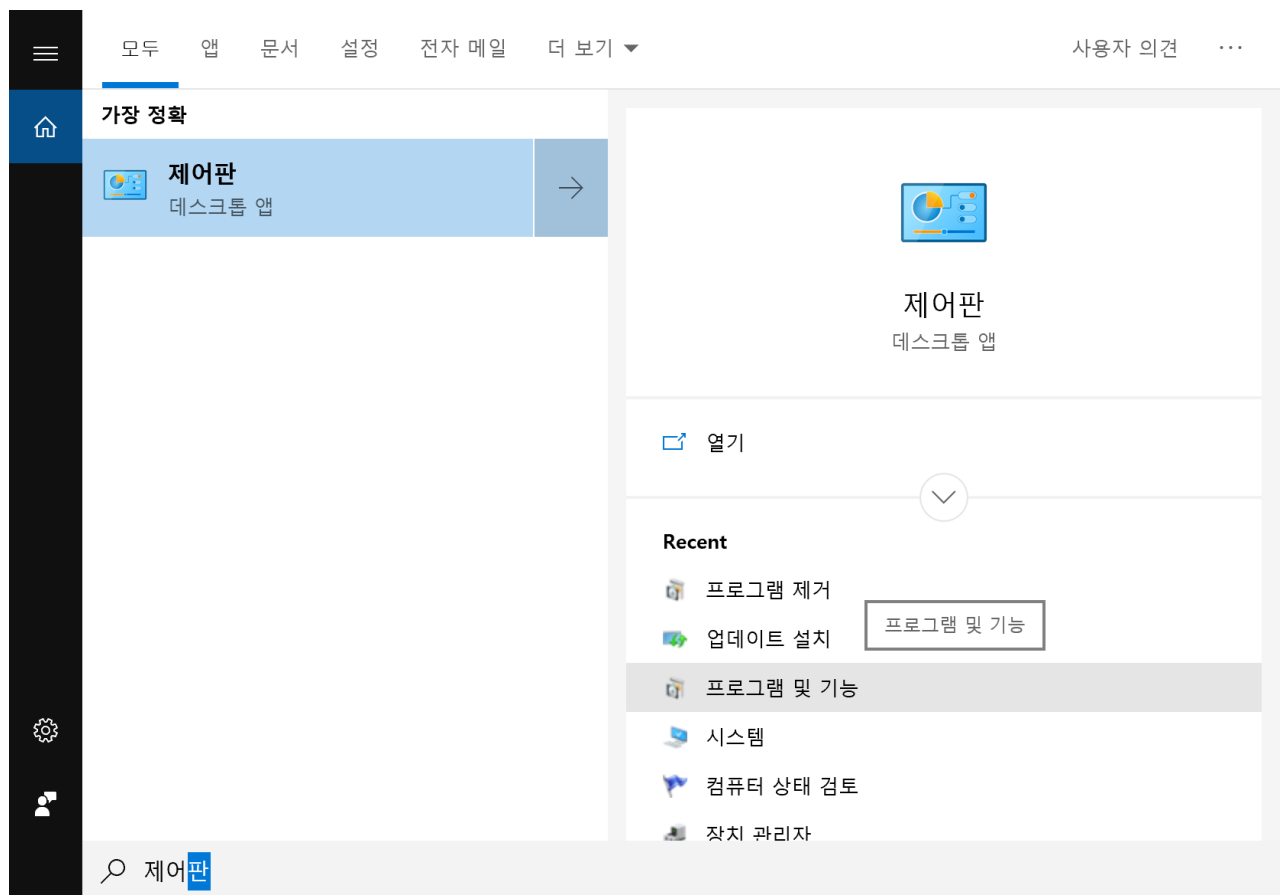
윈도우 10에서 레일스 프로젝트 개발환경 설정하기 (2019년)

Lucius Choi, Founder of RORLAB

윈도우 10에서는 리눅스 배포판을 윈도우 하위시스템(Windows Subsystem for Linux, WSL)으로 설치하여 리눅스 운영체제를 사용할 수 있다. 이제 윈도우 10 에서도 레일스 프로젝트 개발환경을 손쉽게 구축할 수 있게 되었다.

1. 윈도우 10 에 우분투를 설치하자

윈도우 검색창에서 **제어판** 이라고 입력한 후



우측 컬럼에 있는 프로그램 및 기능 항목을 선택한다.

좌측 컬럼에 있는 Windows 기능 켜기/끄기 메뉴를 클릭하며 팝업창이 나타나며 이 때 **Linux용 windows 하위 시스템** 을 체크한 후 확인버튼을 클릭한다.

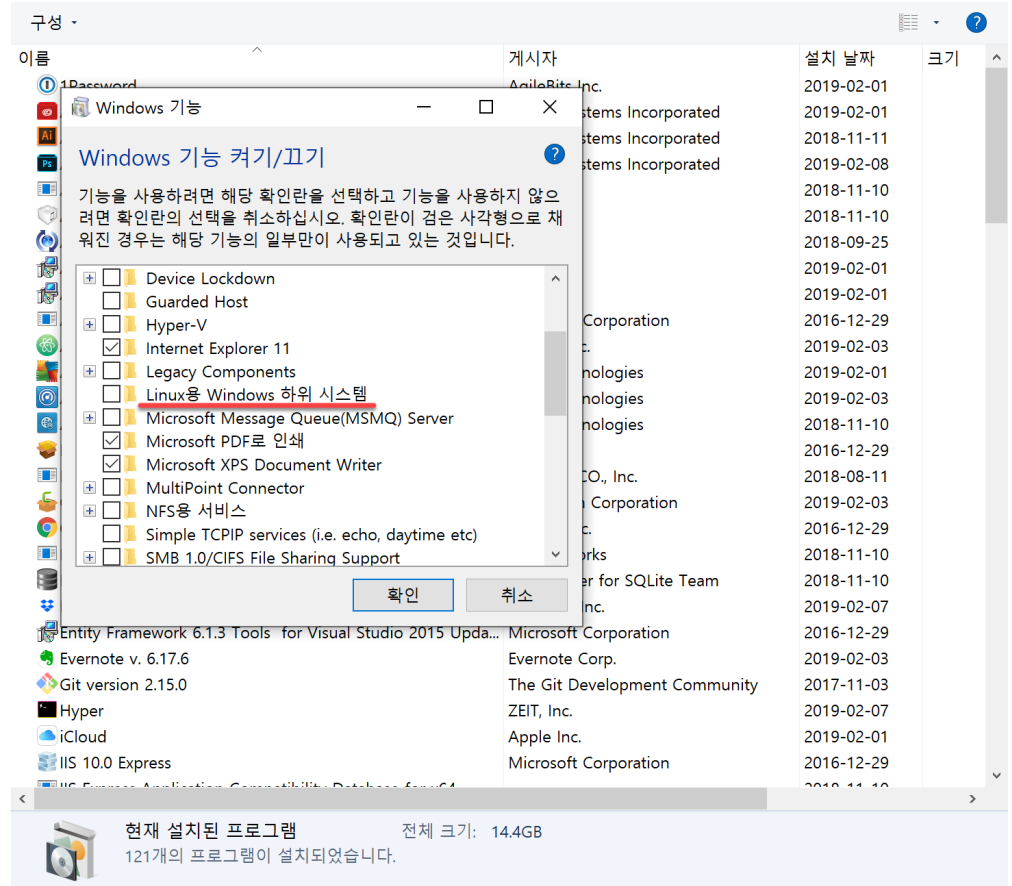
제어판 홈

설치된 업데이트 보기

Windows 기능 켜기/끄기

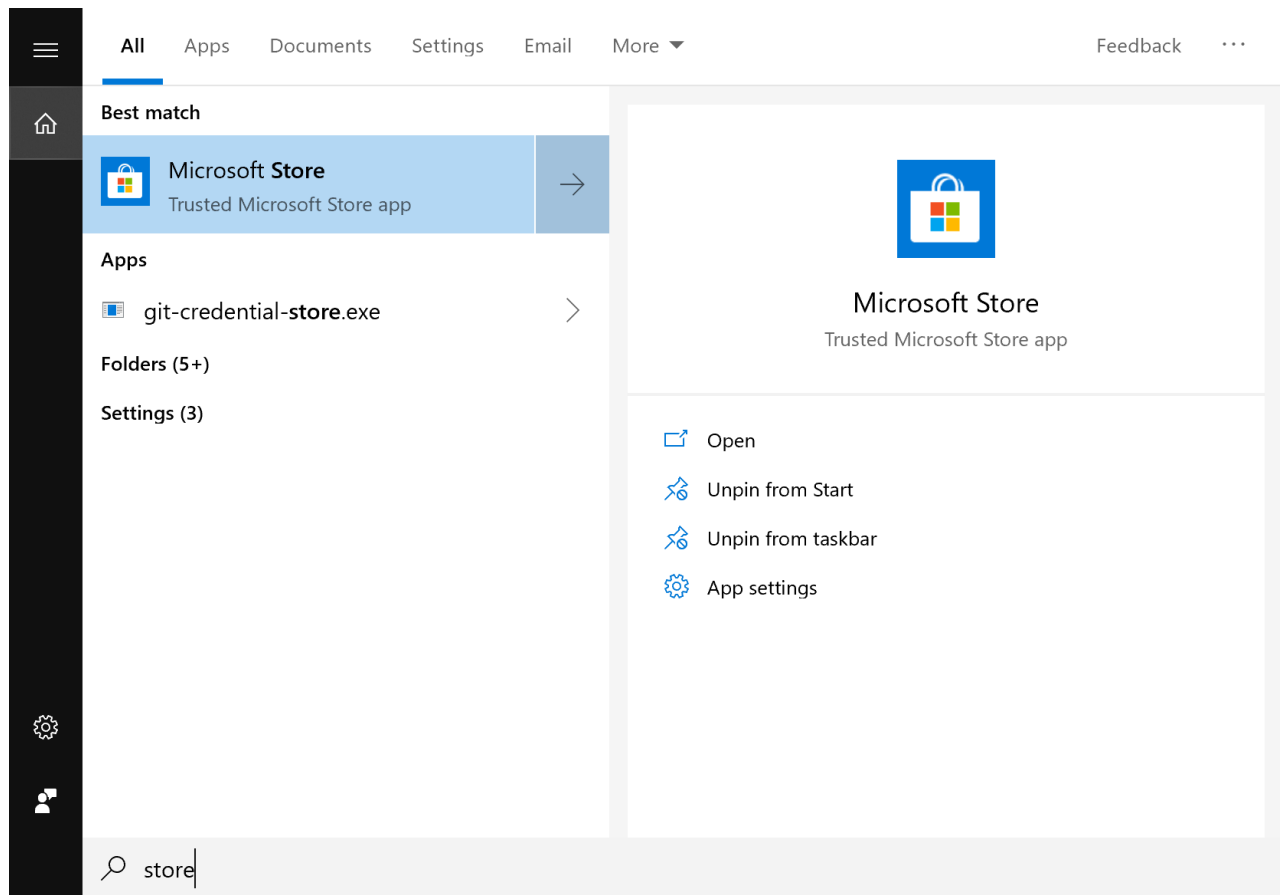
프로그램 제거 또는 변경

프로그램을 제거하려면 목록에서 선택한 후 [제거], [변경] 또는 [복구]를 클릭하십시오.



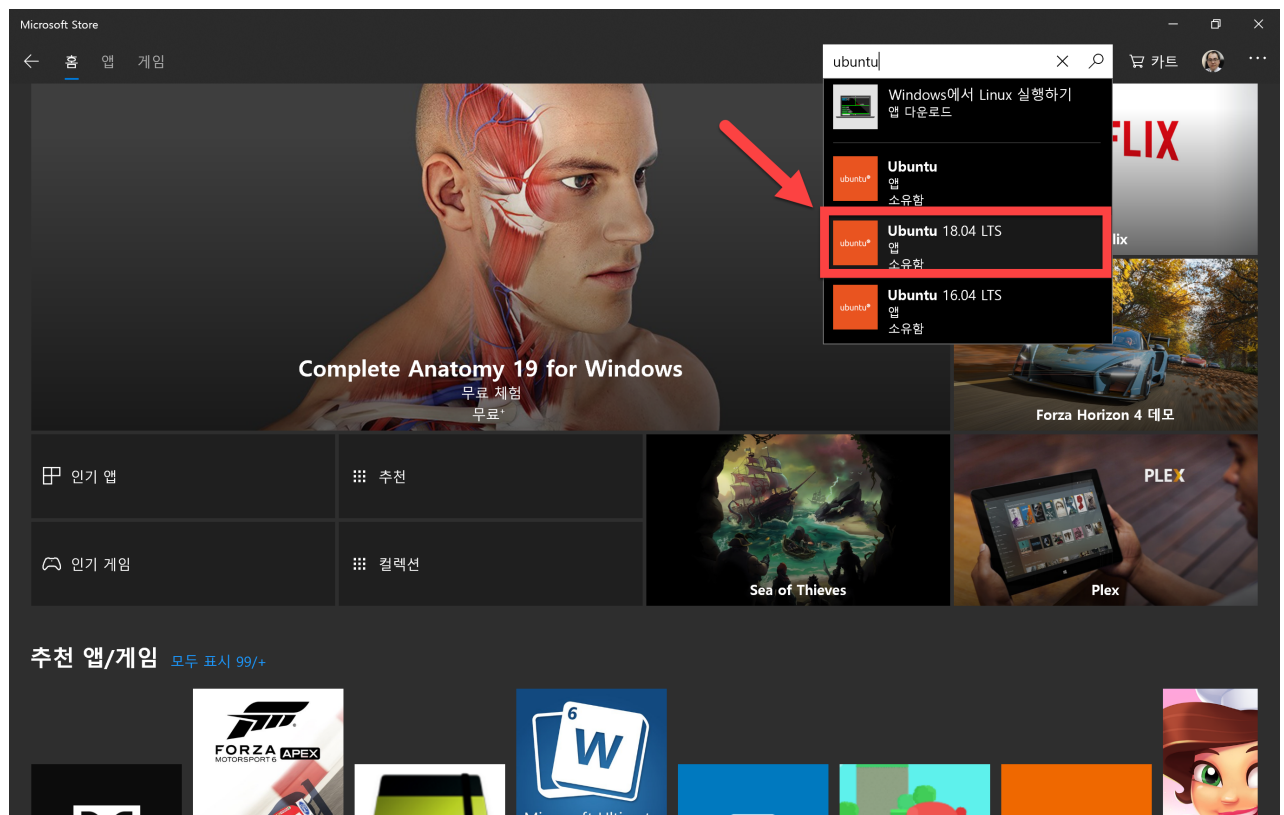
윈도우가 재부팅된 후,

우선 좌측 하단에 있는 윈도우 아이콘을 클릭하거나 키보드에서 좌측 하단에 위치한 윈도우 키를 누르고 **store** 라고 입력하면,

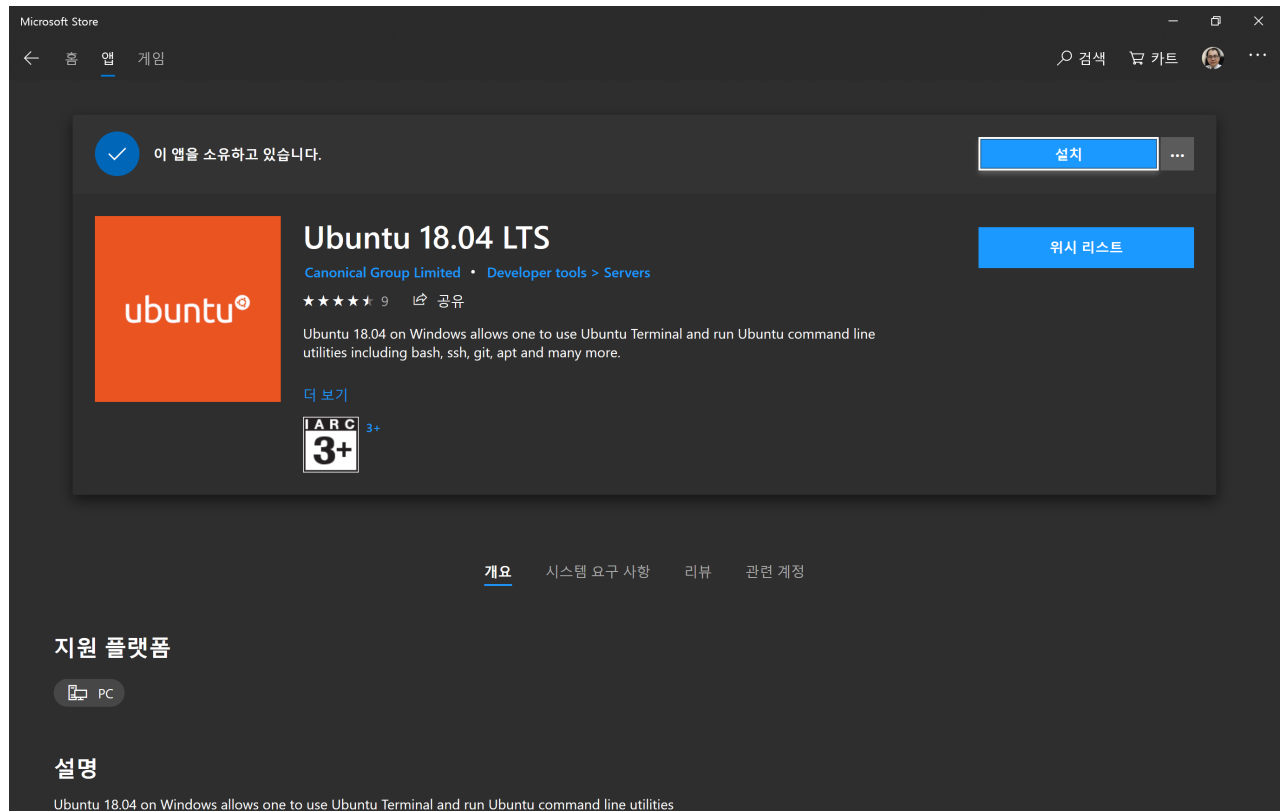


검색 메뉴에 Microsoft Store 앱 항목이 보이게 되는데, 이것을 클릭하여 실행한다.

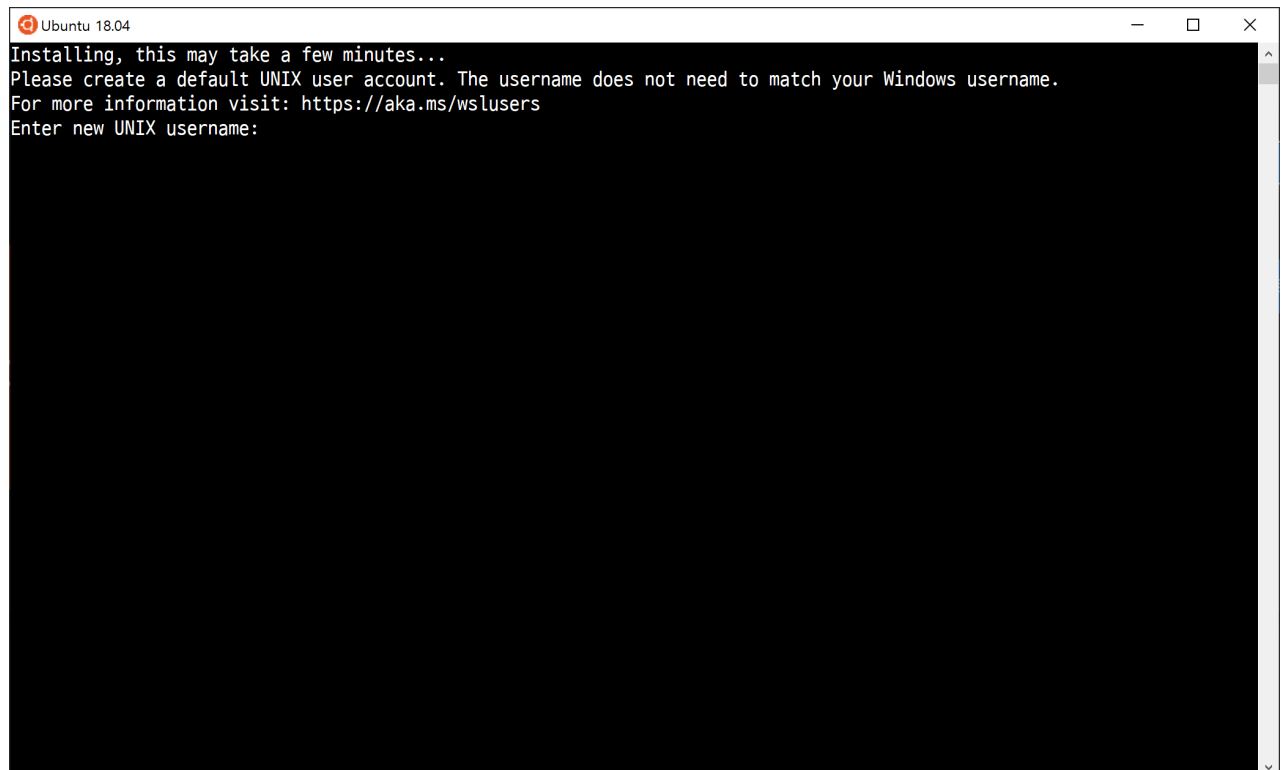
스토어 창의 우측 상단에 있는 검색란에서 **ubuntu** 라고 입력하면 아래와 같이 관련 앱 목록이 보이는데 이 때 18.04 LTS 버전을 선택한다.



다운로드 창의 우측에 있는 설치 버튼을 클릭한다.



다운로드가 완료되면 설치 버튼이 보이게 된다. 이 버튼을 클릭하면 비로서 터미널 창이 보이고 우분투 앱이 설치된다. 이어지는 안내에 따라 진행한다.



슈퍼유저 권한을 가지는 사용자 등록이 완료되면 아래의 내용을 참조하여 레일스 개발환경을 설정한다.

2. 시스템 업데이트 설치하기

```
$ sudo apt update
$ sudo apt list --upgradable # 업데이트 목록 리스트 보기
$ sudo apt upgrade -y
$ sudo apt autoremove -y
```

<https://www.digitalocean.com/community/tutorials/how-to-install-ruby-on-rails-with-rbenv-on-ubuntu-18-04>

3. 운영체제 버전 알아내기

```
$ lsb_release -a
```

```
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 18.04.1 LTS
Release:        18.04
Codename:       bionic
```

4. 레일스 설치에 필요한 의존성 라이브러리 설치하기

```
$ sudo apt install -y autoconf bison build-essential libssl-dev libyaml-dev
libreadline6-dev zlib1g-dev libncurses5-dev libffi-dev libgdbm5 libgdbm-dev
```

5. Zsh 설치하기

<https://medium.com/@vinhp/use-zsh-in-wsl-on-windows-10-5d439a749c4c>

```
$ sudo apt install -y zsh
$ zsh # 설치 옵션 0 을 선택한다.
```

6. oh-my-zsh 설치하기

<https://blog.joaograssi.com/windows-subsystem-for-linux-with-oh-my-zsh-conemu/>

```
$ sh -c "$(curl -fsSL https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/install.sh)"
```

.bashrc 파일을 열고 첫번째 코멘트 직후 아래를 복사해서 붙여넣기 한다.

```
if test -t 1; then
  exec zsh
fi
```

터미널을 종료한 후 WSL(우분투 18.04) 을다시 실행하면 zsh 실행 터미널이 보이게 된다.

7. rbenv 설치하기

```
$ cd
$ git clone https://github.com/rbenv/rbenv.git ~/.rbenv
$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.zshrc
$ echo 'eval "$(rbenv init -)"' >> ~/.zshrc
$ exec $SHELL

$ git clone https://github.com/rbenv/ruby-build.git ~/.rbenv/plugins/ruby-build
$ echo 'export PATH="$HOME/.rbenv/plugins/ruby-build/bin:$PATH"' >> ~/.zshrc
$ exec $SHELL

$ rbenv install 2.6.1
$ rbenv global 2.6.1
$ ruby -v
```

8. 줌 설치시 옵션추가

~/.gemrc 파일을 생성하고 아래와 같은 옵션을 추가한다.

```
$ echo "gem: --no-document" >> ~/.gemrc
$ gem install bundler
```

이것은 향후 줌을 설치할 때 문서 파일을 제외하기 위한 조치이다.

9. Git 환경설정하기

```
$ git config --global color.ui true
$ git config --global user.name "YOUR NAME"
$ git config --global user.email "YOUR@EMAIL.com"
$ ssh-keygen -t rsa -b 4096 -C "YOUR@EMAIL.com"
```

단축키 설정하기

```
$ git config --global alias.co checkout
$ git config --global alias.br branch
$ git config --global alias.ci commit
$ git config --global alias.st status
$ git config --global alias.unstage 'reset HEAD --'
$ git config --global alias.last 'log -1 HEAD'
```

10. Github 에 ssh 공개키 등록하기

생성된 ssh 공개키를 복사한다.

```
$ cat ~/.ssh/id_rsa.pub
```

자신의 github 계정으로 로그인 한 후 설정으로 이동하여 ssh 키를 등록한다. 이제 제대로 설정이 되었는지 확인하기 위해 아래와 같이 쉘명령을 실행한다.

```
$ ssh -T git@github.com
```

11. Nodejs 설치하기

```
$ curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -  
$ sudo apt install -y nodejs
```

12. Yarn(자바스크립트 패키지 매니저) 설치하기

```
$ curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -  
$ echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee  
/etc/apt/sources.list.d/yarn.list  
$ sudo apt update && sudo apt install yarn
```

13. ImageMagick 설치하기

```
$ sudo apt install -y libmagickwand-dev imagemagick
```

14. Sqlite3 설치하기

```
$ sudo apt install -y sqlite3 libsqlite3-dev  
$ sqlite3
```

15. MySQL 설치하기

<https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-18-04>

```
$ sudo apt install -y mysql-server mysql-client libmysqlclient-dev  
$ sudo service mysql start  
$ sudo mysql_secure_installation # root 계정에 암호를 설정할 때  
$ sudo mysql
```

- o root@localhost 계정 접속이 안될 때

-> <https://youtu.be/SJm91cvE ks>

```
$ sudo mysql -u root -p  
mysql> use mysql;  
mysql> select user, host, plugin from mysql.user;  
mysql> update user set plugin='mysql_native_password' where user='root';  
mysql> flush privileges;  
mysql> exit;  
$ mysql -uroot
```

- o 시스템 유저를 MySQL 유저로 추가할 경우

```
$ sudo mysql -u root -p  
mysql> use mysql;  
mysql> create user 'system-username'@'localhost' identified by '';  
mysql> grant all privileges on * . * to 'system-username'@'localhost';  
mysql> update user set plugin='auth_socket' where user='system-username';  
mysql> flush privileges;  
mysql> exit;  
$ mysql
```

- 배포용 계정(`deployer`)을 생성한다.

```
$ sudo mysql -u root -p
mysql> use mysql;
mysql> CREATE USER 'deployer'@'localhost' IDENTIFIED BY 'password';
mysql> GRANT ALL PRIVILEGES ON *.* TO 'deployer'@'localhost' WITH GRANT OPTION;
mysql> FLUSH PRIVILEGES;
mysql> exit;
$ mysql -u deployer -p
```

- 한글깨짐현상 방지하기

(<https://nesoy.github.io/articles/2017-05/mysql-UTF8>) `/etc/mysql/my.cnf` 파일 끝에 아래의 내용을 추가한다.

```
[client]
default-character-set=utf8

[mysql]
default-character-set=utf8

[mysqld]
collation-server = utf8_unicode_ci
init-connect='SET NAMES utf8'
character-set-server = utf8
```

추가한 내용을 저장한 후 MySQL 서버를 재시작한다.

```
$ sudo service mysql restart
```

- 윈도우용 MySQL Workbench 설치하기

(<https://dev.mysql.com/downloads/workbench/>)

주의사항 : 서버의 시작과 종료는 WSL 에서 해야 한다.

```
$ sudo service mysql start|stop|restart|status
```

16. PostgreSQL 설치하기(선택사항)

주의사항 : 윈도우용 PostgreSQL 을 설치하면 안된다. 반드시 리눅스용으로 설치한다. 아래의 링크로 접속하면 방법이 잘 소개되어 있다.

- 참고

1. <https://www.postgresql.org/download/linux/ubuntu/>
2. <https://medium.com/@stephanedmonson/solution-for-connecting-postgresql-via-wsl-windows-subsystem-for-linux-ubuntu18-c79940fa5742>
3. <https://github.com/michaeltreat/Windows-Subsystem-For-Linux-Setup-Guide/blob/master/readmes/installs/PostgreSQL.md>

- WSL로 접속한 후

1. `/etc/apt/sources.list.d/pgdg.list` 파일을 생성하고 아래의 내용을 붙여 넣기 한다. (우분투 18.04 기준)

```
deb http://apt.postgresql.org/pub/repos/apt/ bionic-pgdg main
```

2. 저장소 키를 불러와 패키지 목록을 업데이트 한다.

```
$ wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc |  
sudo apt-key add -  
$ sudo apt update
```

3. 배포판 설치하기 (2019년 2월 현재 최신 버전 11)

```
$ sudo apt install -y postgresql-11
```

4. 의존성 라이브러리 설치하기(pg 켜 설치시에 필요함)

```
$ sudo apt install -y libpq-dev
```

- 배포용 사용자 추가하기

(<https://medium.com/coding-blocks/creating-user-database-and-adding-access-on-postgresql-8bfcd2f4a91e>)

```
$ sudo service postgresql start  
$ sudo -u postgres psql  
postgres=# CREATE USER deployer WITH PASSWORD 'yourpass';  
postgres=# ALTER ROLE deployer superuser createrole createdb replication;
```

- 아래와 같은 인증오류가 발생할 때

```
FATAL: Peer authentication failed for user "deployer"
```

해결방법: <https://gist.github.com/AtulKsol/4470d377b448e56468baef85af7fd614>

- `pg_hba.conf` 파일의 위치 확인하기(hba: host-based authentication)

: db 에 접속해서 아래명령어를 입력해서 확인할 수 있다.

```
$ psql -h localhost -U postgres  
postgres=# SHOW hba_file;
```

- postgres 계정 암호를 지정한다.

```
$ sudo -u postgres psql  
postgres=# alter user postgres password 'password';  
postgres=# \q
```

- SQL 쉘 접속하기

<http://postgresguide.com/utilities/psql.html>

```
$ psql -h localhost -U postgres [postgres] [-p 5432]
```

o pgAdmin 4 (Windows) 설치하기

- 다운로드 : <https://www.pgadmin.org/download/pgadmin-4-windows/>
- 설치후 서버 추가하기 : <https://stackoverflow.com/a/54192456>
- postgres 계정의 비밀번호 인증오류시 : <https://stackoverflow.com/a/7696398>
- 주의사항 : 서버의 기동은 WSL 에서 해야 한다.

```
$ sudo service postgresql [start|stop|restart|reload|force-reload|status]
```

o PostgreSQL 완전히 제거하기

PostgreSQL을 깨끗하게 재설치할 때는 아래와 같이 실행한다.

```
$ sudo apt --purge remove postgresql/*
```

15. 윈도우 파일 시스템에 접근하기 (<https://code.apptilus.com/posts/tools/windows-subsystem-linux>)

WSL 내에서 윈도우 파일시스템에 접근하기 위해서 다음과 같이 필요한 폴더를 마운트한다.

```
# D드라이브 하위의 workspace 폴더에 접근하기
$ cd /mnt/d/workspace
```

매번 윈도우 내부의 작업 디렉토리로 이동하기 위해 위와 같이 명령어를 입력하는 것은 매우 귀찮은 작업이다. 따라서, 심볼릭 링크를 이용하여 마치 WSL 내부의 디렉토리를 이용하듯 손쉽게 윈도우 폴더에 접근하도록 한다.

```
# symbolic link 사용
$ ln -s "/mnt/d/workspace" /home/<my-wsl-username>/workspace
```

위와 같이 심볼릭 링크를 구성하면 WSL에서 `cd workspace` 명령만으로 간단하게 윈도우의 Workspace 폴더에 접근할 수 있게 된다.

16. Rails 설치하기

```
$ gem install rails
```

설치시에 나타나는 안내문:

```
HEADS UP! i18n 1.1 changed fallbacks to exclude default locale.
But that may break your application.

Please check your Rails app for 'config.i18n.fallbacks = true'.
If you're using I18n (>= 1.1.0) and Rails (< 5.2.2), this should be
'config.i18n.fallbacks = [I18n.default_locale]'.
If not, fallbacks will be broken in your app by I18n 1.1.x.
```

레일스 서버 실행시에는 가급적 `bin` 디렉토리에 있는 실행파일을 사용할 것을 권한다.

```
$ bin/rails s
$ bin/rails db:create
$ bin/rails db:migrate
$ bin/rails g scaffold ....
```