HACK YOUR LIFE!

멋쟁이 사자처럼

Vagrant 로 로컬 개발 환경 만들기

for windows OS

멋쟁이 사자처럼 at 평택대학교

시작 전 해야 할 일

◥ VirtualBox 설치

◥ Vagrant 설치

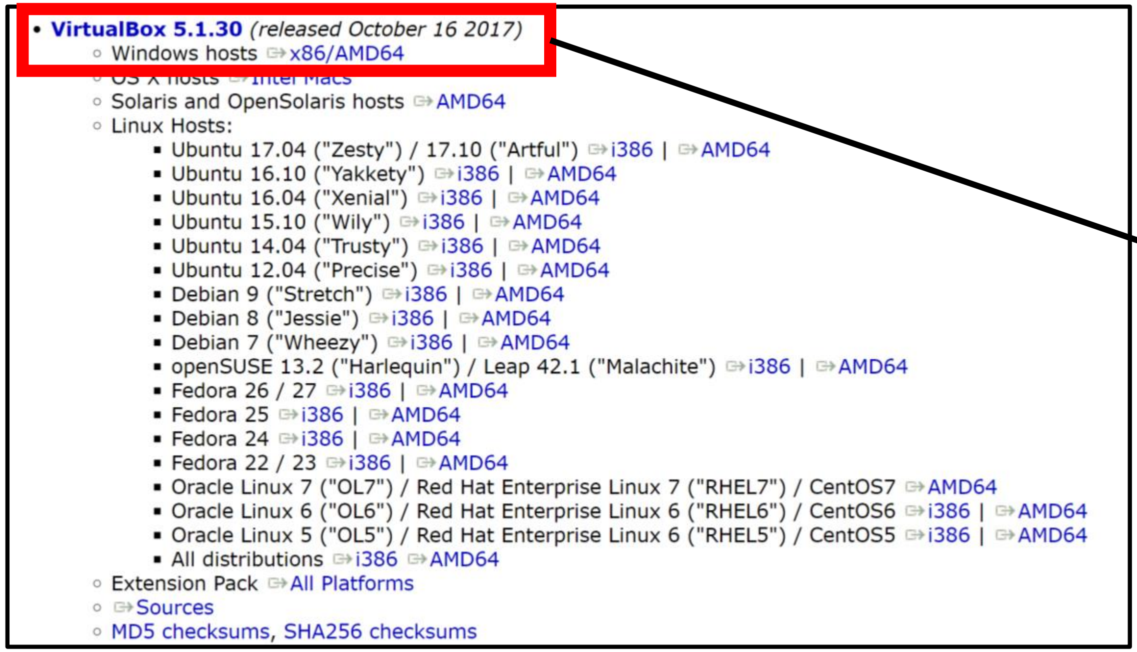
◥ Git 설치

- 설치 시 모든 설정은 초기 설정 유지!

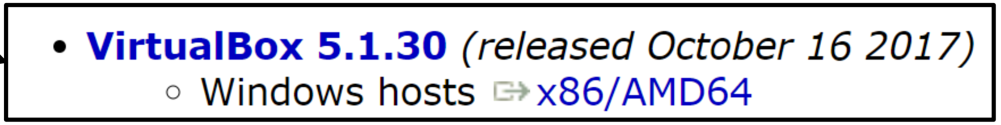
VirtualBox 설치

https://www.virtualbox.org/wiki/Download_Old_Builds_5_1

- 위 링크로 접속하여 VirtualBox 5.1.30 version 설치



- VirtualBox 5.1.30 *(released October 16 2017)*
  - Windows hosts ⇨x86/AMD64
  - OS X hosts ⇨Intel Macs
  - Solaris and OpenSolaris hosts ⇨AMD64
  - Linux Hosts:
    - Ubuntu 17.04 ("Zesty") / 17.10 ("Artful") ⇨i386 | ⇨AMD64
    - Ubuntu 16.10 ("Yakkety") ⇨i386 | ⇨AMD64
    - Ubuntu 16.04 ("Xenial") ⇨i386 | ⇨AMD64
    - Ubuntu 15.10 ("Wily") ⇨i386 | ⇨AMD64
    - Ubuntu 14.04 ("Trusty") ⇨i386 | ⇨AMD64
    - Ubuntu 12.04 ("Precise") ⇨i386 | ⇨AMD64
    - Debian 9 ("Stretch") ⇨i386 | ⇨AMD64
    - Debian 8 ("Jessie") ⇨i386 | ⇨AMD64
    - Debian 7 ("Wheezy") ⇨i386 | ⇨AMD64
    - openSUSE 13.2 ("Harlequin") / Leap 42.1 ("Malachite") ⇨i386 | ⇨AMD64
    - Fedora 26 / 27 ⇨i386 | ⇨AMD64
    - Fedora 25 ⇨i386 | ⇨AMD64
    - Fedora 24 ⇨i386 | ⇨AMD64
    - Fedora 22 / 23 ⇨i386 | ⇨AMD64
    - Oracle Linux 7 ("OL7") / Red Hat Enterprise Linux 7 ("RHEL7") / CentOS7 ⇨AMD64
    - Oracle Linux 6 ("OL6") / Red Hat Enterprise Linux 6 ("RHEL6") / CentOS6 ⇨i386 | ⇨AMD64
    - Oracle Linux 5 ("OL5") / Red Hat Enterprise Linux 6 ("RHEL5") / CentOS5 ⇨i386 | ⇨AMD64
    - All distributions ⇨i386 ⇨AMD64
  - Extension Pack ⇨All Platforms
  - ⇨Sources
  - MD5 checksums, SHA256 checksums



- VirtualBox 5.1.30 *(released October 16 2017)*
  - Windows hosts ⇨x86/AMD64

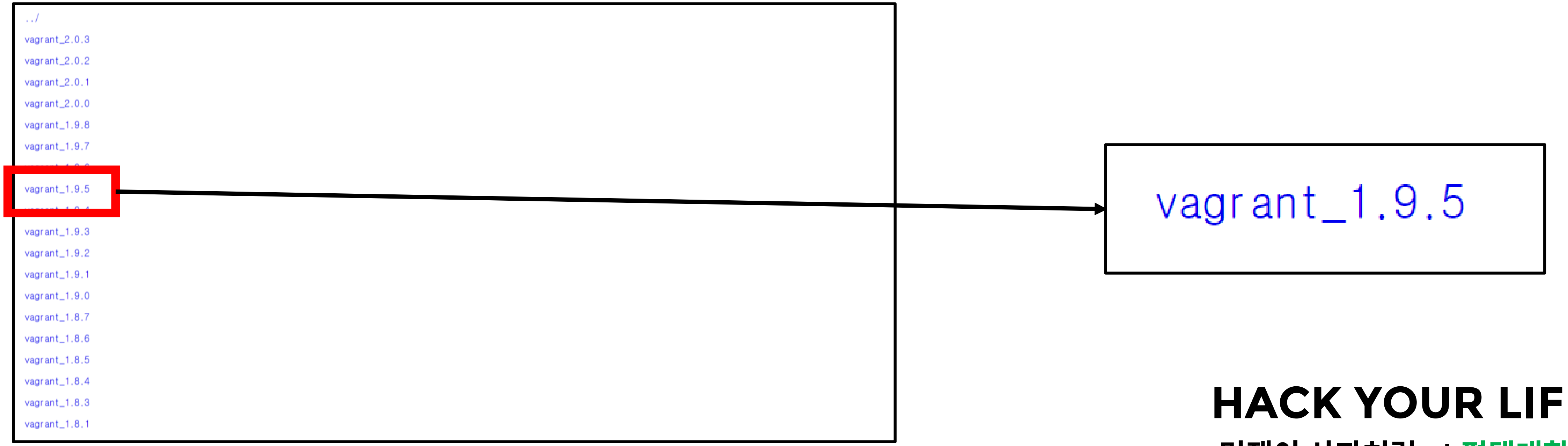
HACK YOUR LIFE!

멋쟁이 사자처럼 at 평택대학교

Vagrant 설치

- https://releases.hashicorp.com/vagrant/?_ga=2.237010176.1475687836.1513147132-756484628.1513147132

  - 위 링크로 접속하여 Vagrant_1.9.5 version 설치



```
../
vagrant_2.0.3
vagrant_2.0.2
vagrant_2.0.1
vagrant_2.0.0
vagrant_1.9.8
vagrant_1.9.7
vagrant_1.9.6
vagrant_1.9.5
vagrant_1.9.4
vagrant_1.9.3
vagrant_1.9.2
vagrant_1.9.1
vagrant_1.9.0
vagrant_1.8.7
vagrant_1.8.6
vagrant_1.8.5
vagrant_1.8.4
vagrant_1.8.3
vagrant_1.8.1
```

vagrant_1.9.5

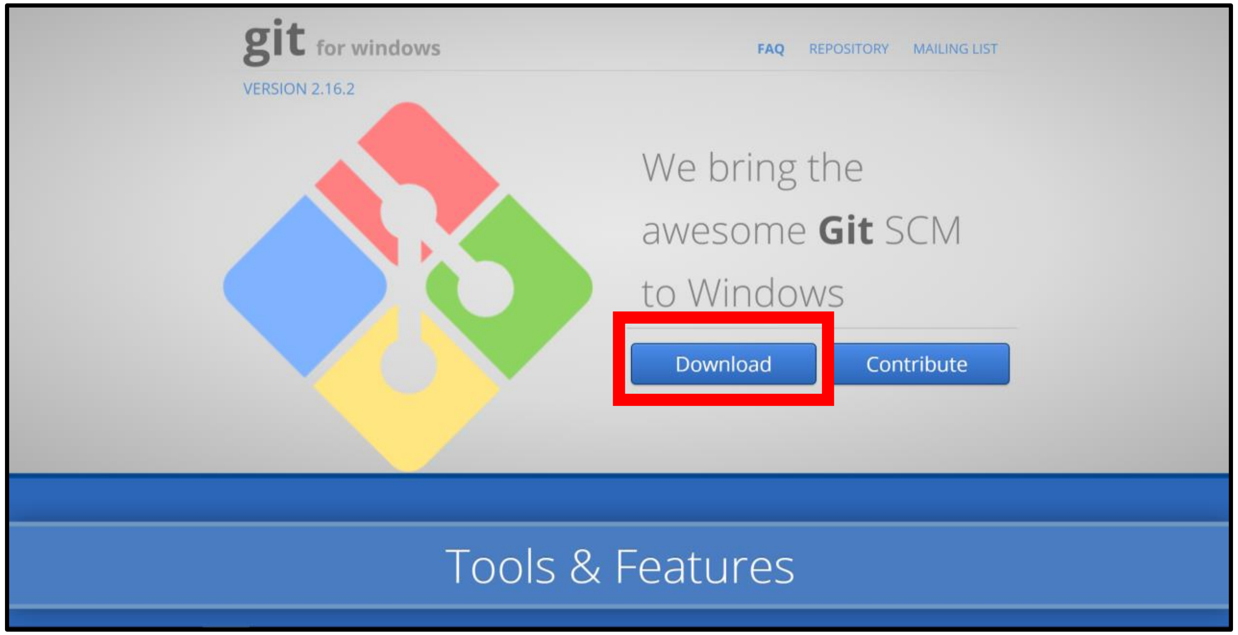
HACK YOUR LIFE!

멋쟁이 사자처럼 at 평택대학교

Git for window 설치

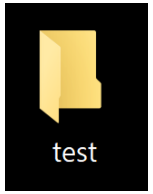
https://gitforwindows.org/

- 위 링크로 접속하여 현재 버전 설치



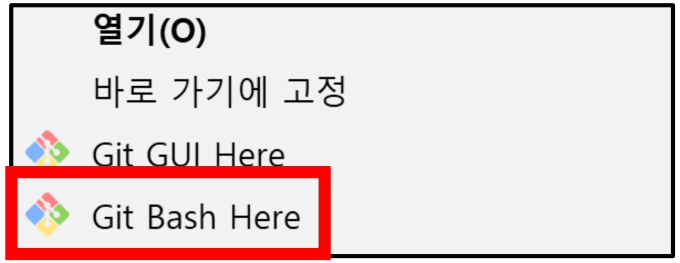
가상 환경 설정하기

※지금부터는 2번째 강의의 “github 링크 자료” 를 중심으로 보며 참조하세요



1. 원하는 위치에 원하는 이름의 폴더를 만듭니다.

- 한글 이름도 상관 없는 것 같지만, 혹시 모르니 영어 이름으로 만들기를 권장합니다!



열기(O)

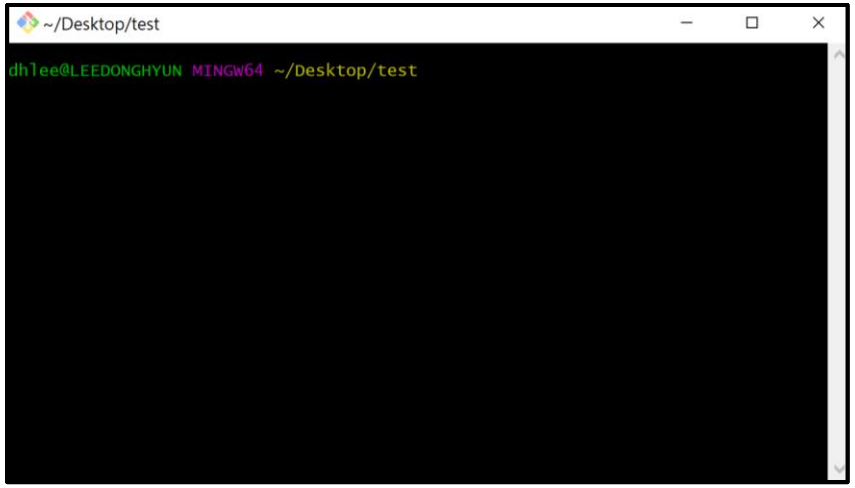
바로 가기에 고정

Git GUI Here

Git Bash Here

2. 폴더를 우 클릭 후 git Bash Here을 눌러줍시다!

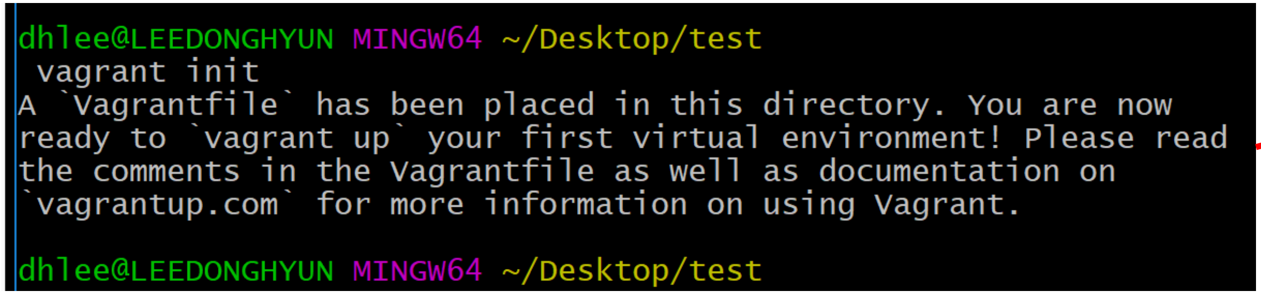
가상 환경 설정하기



```
dhlee@LEEDONGHYUN MINGW64 ~/Desktop/test
```

3. 요런게 나왔을텐데

vagrant init 명령어를 입력 후 Enter!



```
dhlee@LEEDONGHYUN MINGW64 ~/Desktop/test
 vagrant init
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.

dhlee@LEEDONGHYUN MINGW64 ~/Desktop/test
```

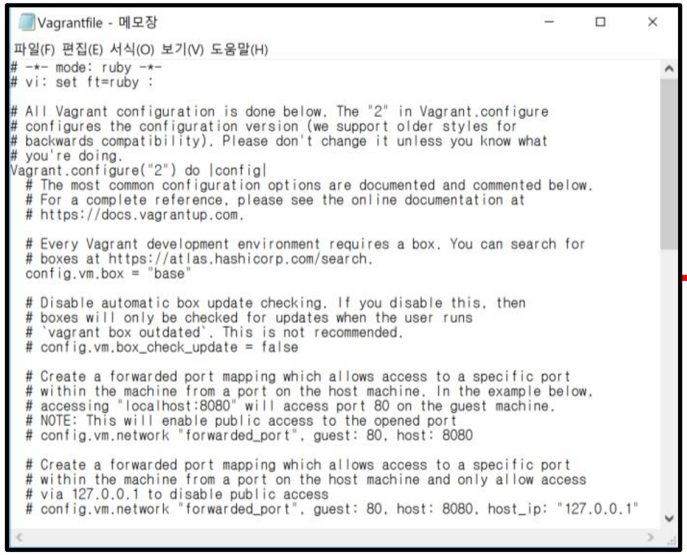
이런 결과가 나왔다면 성공!

해당 폴더로 들어가 봅시다.

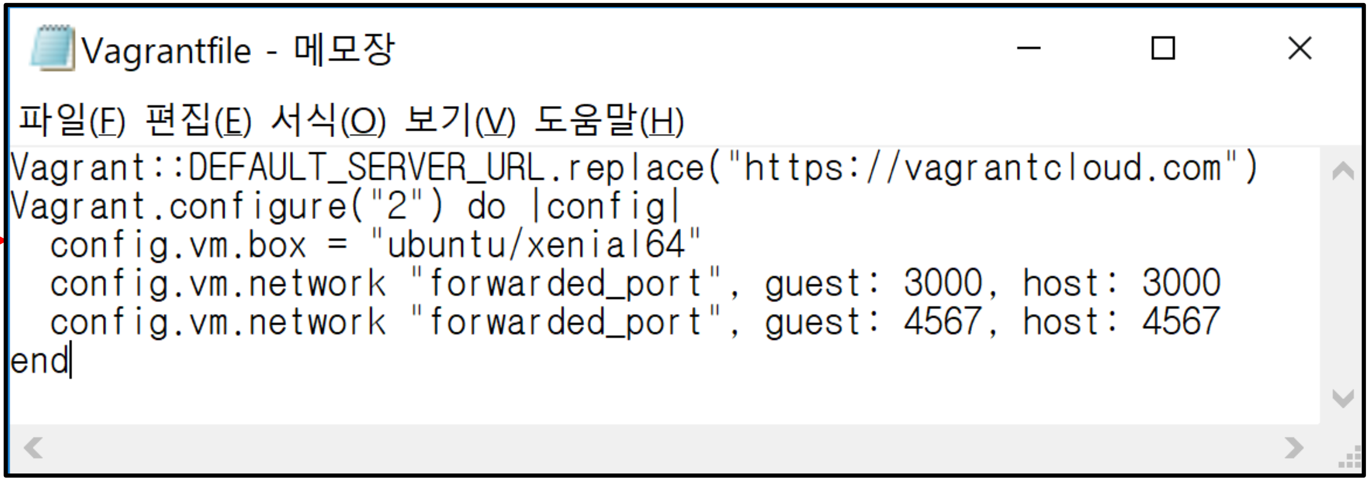
가상 환경 설정하기

Vagrantfile

4. Vagrantfile을 텍스트 편집기로 열고 내용을 바꾸어 저장합니다.



```
# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://atlas.hashicorp.com/search.
  config.vm.box = "base"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example below,
  # accessing "localhost:8080" will access port 80 on the guest machine.
  # NOTE: This will enable public access to the opened port
  # config.vm.network "forwarded_port", guest: 80, host: 8080

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine and only allow access
  # via 127.0.0.1 to disable public access
  # config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"
```



```
Vagrant::DEFAULT_SERVER_URL.replace("https://vagrantcloud.com")
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/xenial64"
  config.vm.network "forwarded_port", guest: 3000, host: 3000
  config.vm.network "forwarded_port", guest: 4567, host: 4567
end
```

- 텍스트를 편집 할 수 있다면 어떤 프로그램으로 수정해도 관계는 없지만

Atom이나 sublime과 같은 text-editer를 설치하여 사용할 것을 권장합니다!

HACK YOUR LIFE!
멋쟁이 사자처럼 at 평택대학교

가상 환경 만들기

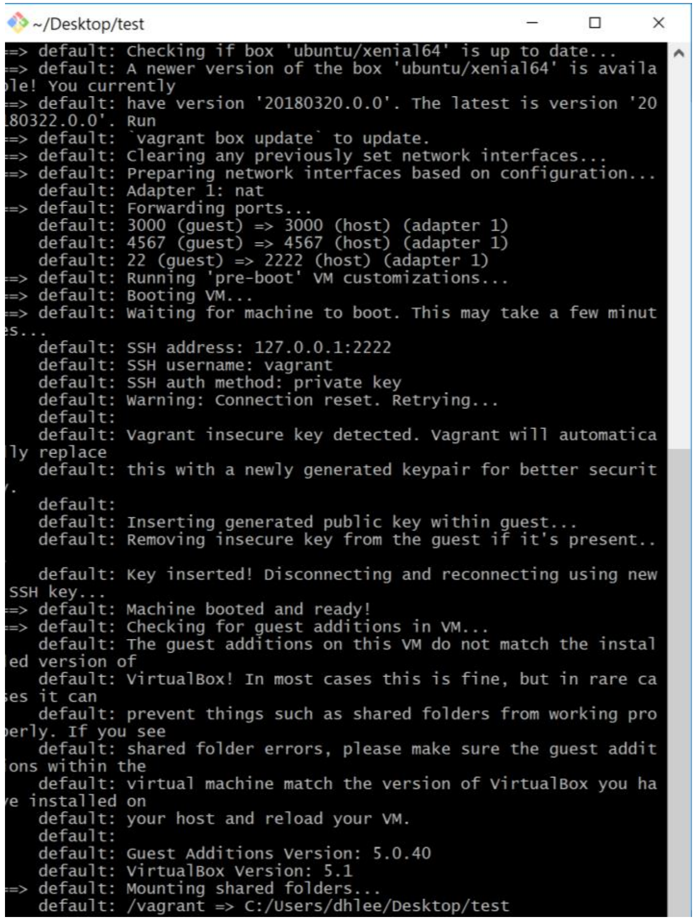
5. 다시 bash로 돌아와

Vagrant ssh 명령어를 입력 후 Enter!

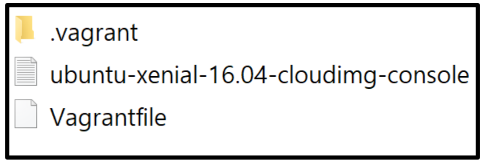
- 우측과 같은 결과들이 나오지 않았다면,

사전 설치 파일들이 정확히 설치 되었는지 확인!

- 환경에 따라 시간이 많이 소요될 수 있습니다.


~/Desktop/test

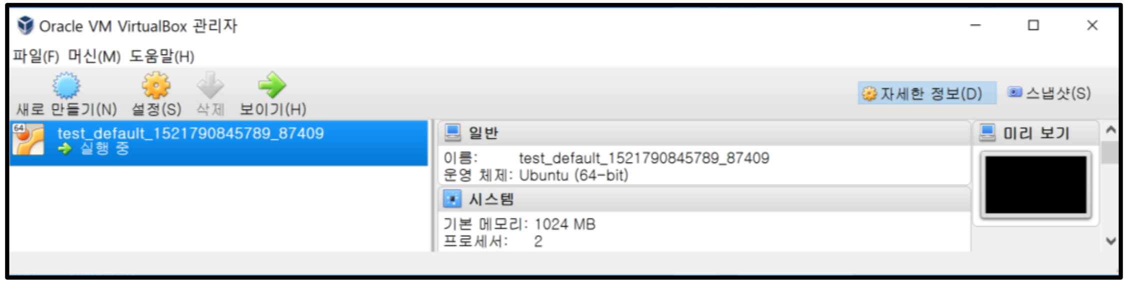
```
=> default: Checking if box 'ubuntu/xenial64' is up to date...
=> default: A newer version of the box 'ubuntu/xenial64' is availa
le! You currently
=> default: have version '20180320.0.0'. The latest is version '20
180322.0.0'. Run
=> default: `vagrant box update` to update.
=> default: Clearing any previously set network interfaces...
=> default: Preparing network interfaces based on configuration...
   default: Adapter 1: nat
=> default: Forwarding ports...
   default: 3000 (guest) => 3000 (host) (adapter 1)
   default: 4567 (guest) => 4567 (host) (adapter 1)
   default: 22 (guest) => 2222 (host) (adapter 1)
=> default: Running 'pre-boot' VM customizations...
=> default: Booting VM...
=> default: Waiting for machine to boot. This may take a few minut
es...
   default: SSH address: 127.0.0.1:2222
   default: SSH username: vagrant
   default: SSH auth method: private key
   default: Warning: Connection reset. Retrying...
   default:
   default: Vagrant insecure key detected. Vagrant will automatica
ly replace
   default: this with a newly generated keypair for better securit
y.
   default:
   default: Inserting generated public key within guest...
   default: Removing insecure key from the guest if it's present..

   default: Key inserted! Disconnecting and reconnecting using new
 SSH key...
=> default: Machine booted and ready!
=> default: Checking for guest additions in VM...
   default: The guest additions on this VM do not match the instal
led version of
   default: VirtualBox! In most cases this is fine, but in rare ca
ses it can
   default: prevent things such as shared folders from working pro
perly. If you see
   default: shared folder errors, please make sure the guest addit
ions within the
   default: virtual machine match the version of VirtualBox you ha
ve installed on
   default: your host and reload your VM.
   default:
   default: Guest Additions Version: 5.0.40
   default: VirtualBox Version: 5.1
=> default: Mounting shared folders...
   default: /vagrant => C:/Users/dhlee/Desktop/test
```


.vagrant

ubuntu-xenial-16.04-cloudimg-console

Vagrantfile

-명령어 이후 생성된 폴더와 파일



test_default_1521790845789_87409
실행 중

이름: test_default_1521790845789_87409
운영 체제: Ubuntu (64-bit)

기본 메모리: 1024 MB
프로세서: 2

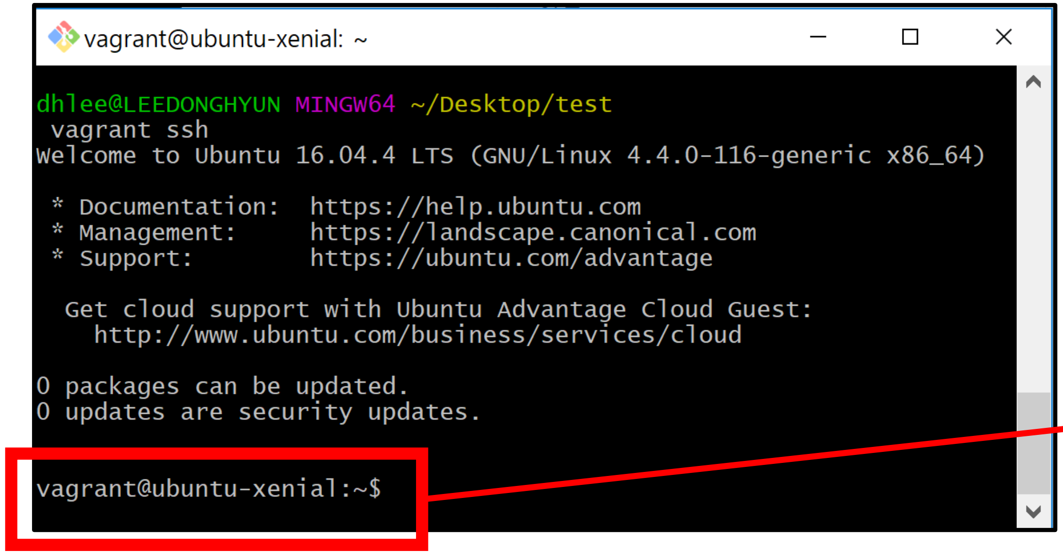
-최종적으로 만들어진 가상 환경은

VirtualBox 관리자에서 확인 가능합니다.

HACK YOUR LIFE!

멋쟁이 사자처럼 at 평택대학교

가상 환경 접속하기



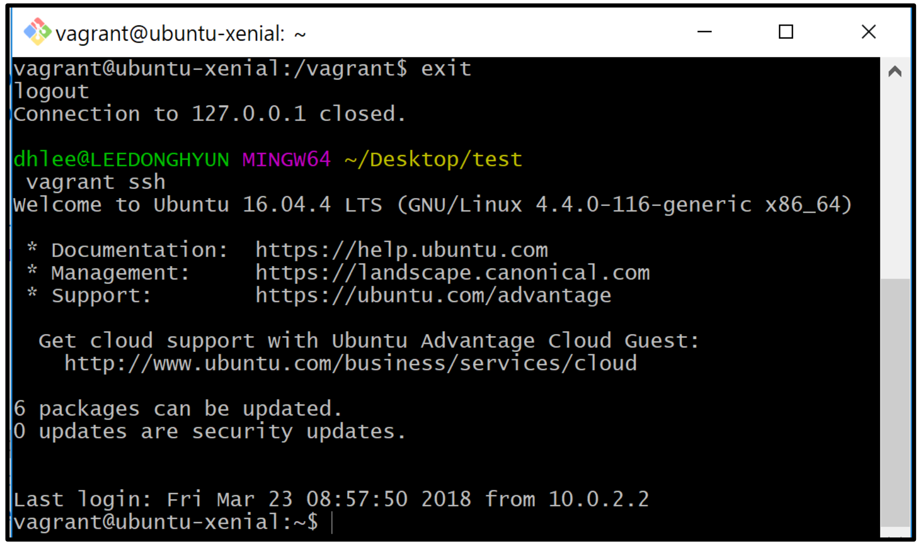
```
dhlee@LEEDONGHYUN MINGW64 ~/Desktop/test
 vagrant ssh
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-116-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

vagrant@ubuntu-xenial:~$
```

6. 이어서 vagrant ssh 명령어를 입력 후 Enter!

-이 상태가 가상 머신에 접속되었음을 의미합니다.

- 위와 같은 결과가 나오지 않았다면, 가상 환경이 정상적으로 만들어 졌는지 확인합시다!

Ruby 개발환경 만들기

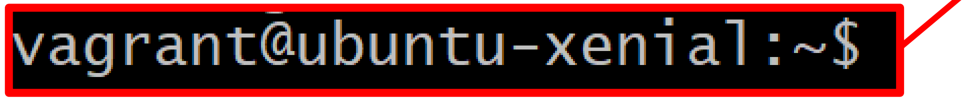


```
vagrant@ubuntu-xenial:/vagrant$ exit
logout
Connection to 127.0.0.1 closed.

dhlee@LEEDONGHYUN MINGW64 ~/Desktop/test
 vagrant ssh
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-116-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

6 packages can be updated.
0 updates are security updates.


Last login: Fri Mar 23 08:57:50 2018 from 10.0.2.2
vagrant@ubuntu-xenial:~$
```

8. 이어서 할 경우 :

가상 환경에서 설치 되어야 하기 때문에

"exit" 명령어로 로그아웃 후 다시 접속하길 바랍니다.



```
vagrant@ubuntu-xenial:~$
```

- 이와 같은 상태에서 진행되어야 합니다.

Ruby 개발환경 만들기

9. Bash에 순서대로 한줄 한줄 복사 붙여 넣기 후 Enter!

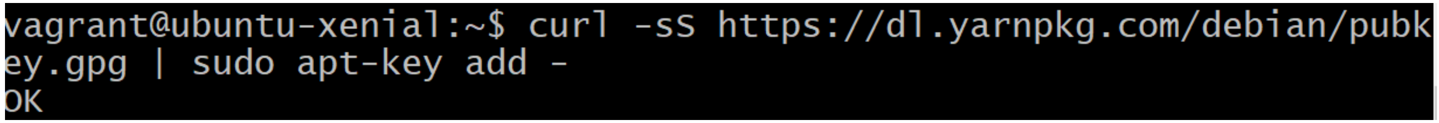
- 지금부터는 명령어 입력 시 결과 확인용으로, 같은 결과가 나왔는지 비교하시면 되겠습니다.

오류가 발생하면 명령어 행을 정확히 입력했는지 확인해봅시다!

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
```

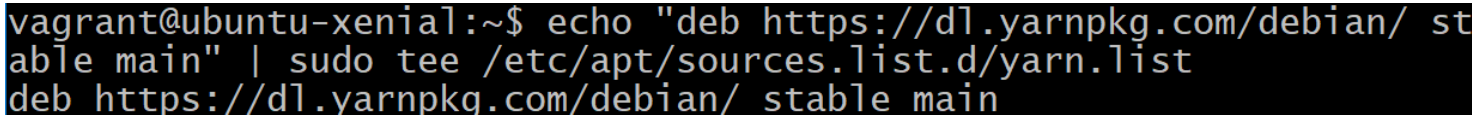
- 이것 저것 진행됩니다. 시간이 많이 소요되면 정상 작동입니다.

```
curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -
```



```
vagrant@ubuntu-xenial:~$ curl -sS https://dl.yarnpkg.com/debian/pubk
ey.gpg | sudo apt-key add -
OK
```

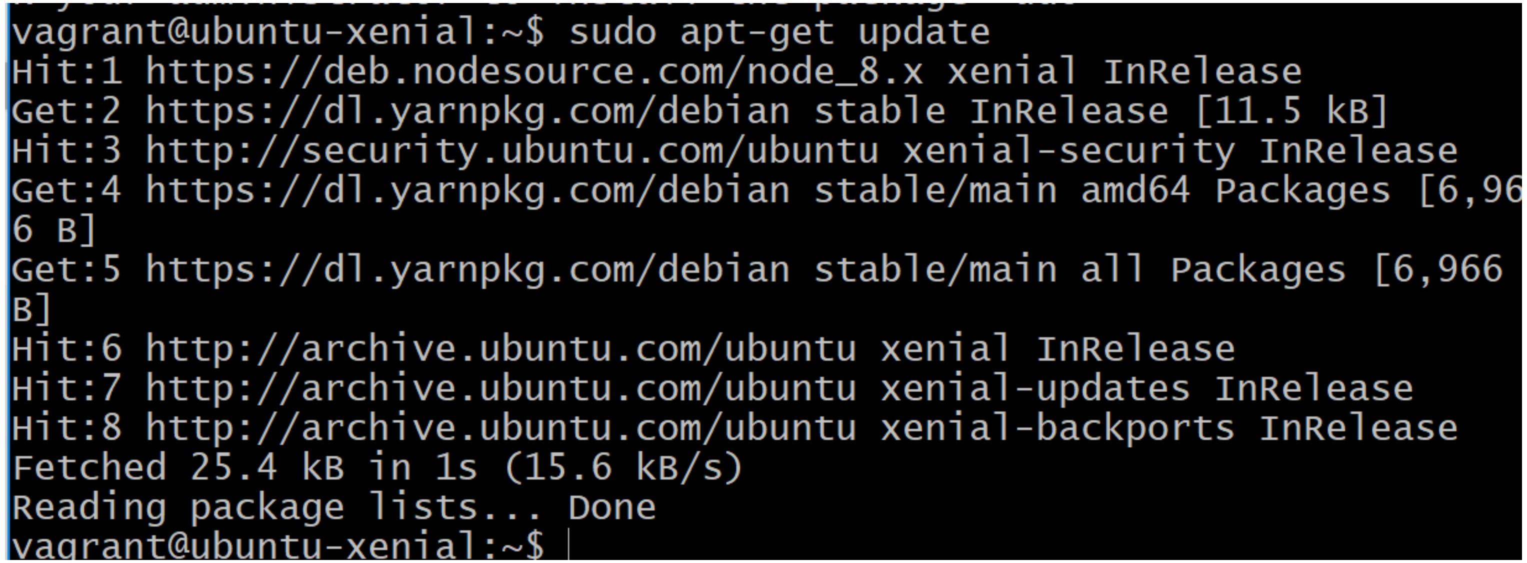
```
echo "deb https://dl.yarnpkg.com/debian/
stable main" | sudo tee /etc/apt/sources.list.d/yarn.list
```



```
vagrant@ubuntu-xenial:~$ echo "deb https://dl.yarnpkg.com/debian/ st
able main" | sudo tee /etc/apt/sources.list.d/yarn.list
deb https://dl.yarnpkg.com/debian/ stable main
```

Ruby 개발환경 만들기

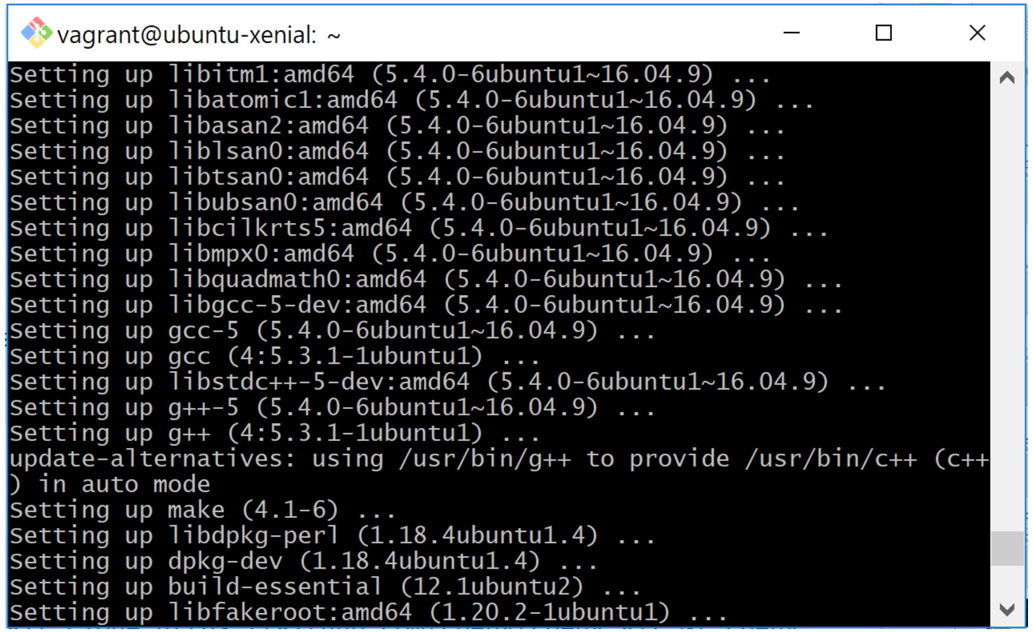
```
sudo apt-get update
```



```
vagrant@ubuntu-xenial:~$ sudo apt-get update
Hit:1 https://deb.nodesource.com/node_8.x xenial InRelease
Get:2 https://dl.yarnpkg.com/debian stable InRelease [11.5 kB]
Hit:3 http://security.ubuntu.com/ubuntu xenial-security InRelease
Get:4 https://dl.yarnpkg.com/debian stable/main amd64 Packages [6,96
6 B]
Get:5 https://dl.yarnpkg.com/debian stable/main all Packages [6,966
B]
Hit:6 http://archive.ubuntu.com/ubuntu xenial InRelease
Hit:7 http://archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:8 http://archive.ubuntu.com/ubuntu xenial-backports InRelease
Fetched 25.4 kB in 1s (15.6 kB/s)
Reading package lists... Done
vagrant@ubuntu-xenial:~$
```

Ruby 개발환경 만들기

```
sudo apt-get install git-core curl zlib1g-dev build-essential
 libssl-dev libreadline-dev libyaml-dev libsqlite3-dev sqlite3
 libxml2-dev libxslt1-dev libcurl4-openssl-dev python-software-properties
 libffi-dev nodejs yarn
```

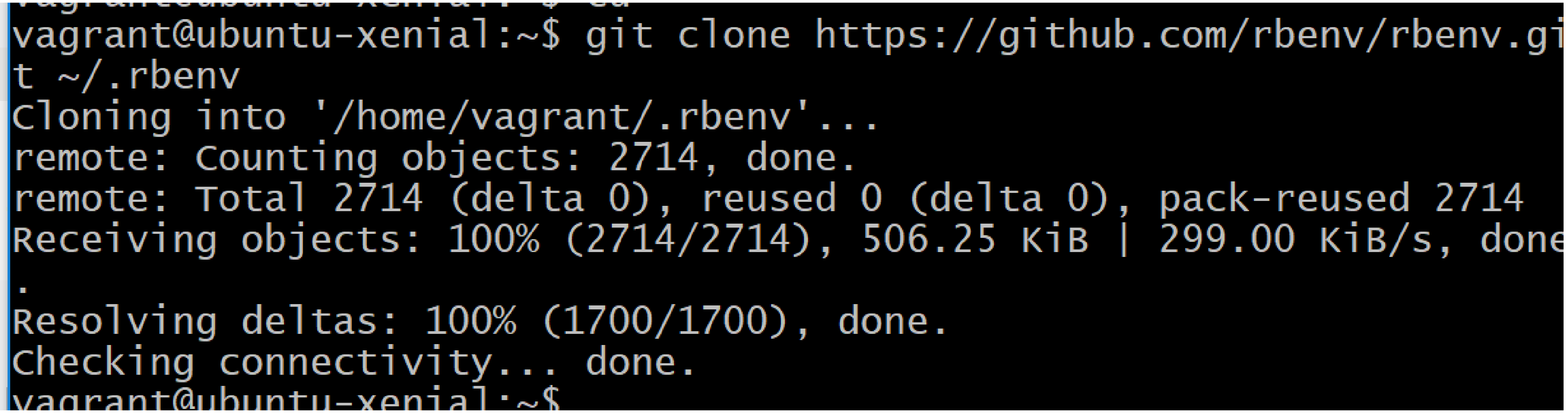


```
Setting up libitm1:amd64 (5.4.0-6ubuntu1~16.04.9) ...
Setting up libatomic1:amd64 (5.4.0-6ubuntu1~16.04.9) ...
Setting up libasan2:amd64 (5.4.0-6ubuntu1~16.04.9) ...
Setting up liblsan0:amd64 (5.4.0-6ubuntu1~16.04.9) ...
Setting up libtsan0:amd64 (5.4.0-6ubuntu1~16.04.9) ...
Setting up libubsan0:amd64 (5.4.0-6ubuntu1~16.04.9) ...
Setting up libcilkrts5:amd64 (5.4.0-6ubuntu1~16.04.9) ...
Setting up libmpx0:amd64 (5.4.0-6ubuntu1~16.04.9) ...
Setting up libquadmath0:amd64 (5.4.0-6ubuntu1~16.04.9) ...
Setting up libgcc-5-dev:amd64 (5.4.0-6ubuntu1~16.04.9) ...
Setting up gcc-5 (5.4.0-6ubuntu1~16.04.9) ...
Setting up gcc (4:5.3.1-1ubuntu1) ...
Setting up libstdc++-5-dev:amd64 (5.4.0-6ubuntu1~16.04.9) ...
Setting up g++-5 (5.4.0-6ubuntu1~16.04.9) ...
Setting up g++ (4:5.3.1-1ubuntu1) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++
) in auto mode
Setting up make (4.1-6) ...
Setting up libdpkg-perl (1.18.4ubuntu1.4) ...
Setting up dpkg-dev (1.18.4ubuntu1.4) ...
Setting up build-essential (12.1ubuntu2) ...
Setting up libfakeroot:amd64 (1.20.2-1ubuntu1) ...
```

- 이것 저것 자동으로 진행되는데 시간이 꽤 걸립니다.

Ruby 개발환경 만들기

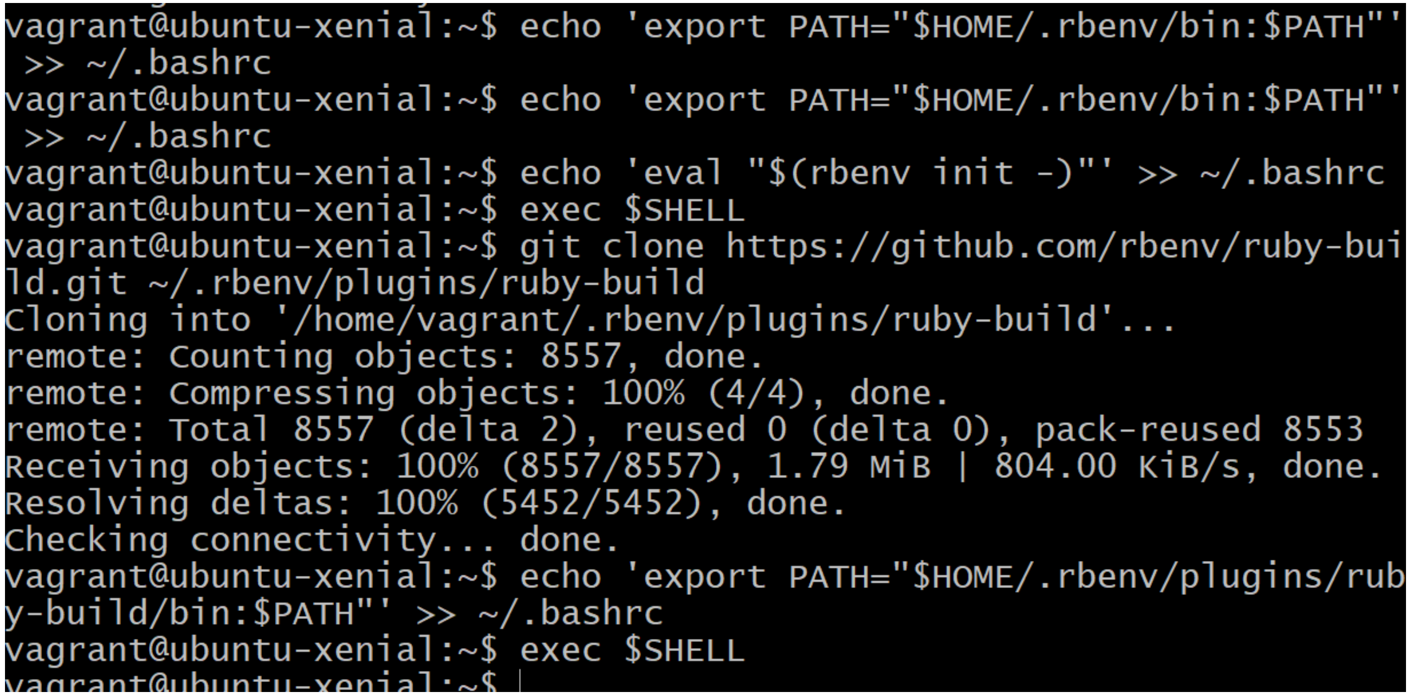
```
git clone https://github.com/rbenv/rbenv.git ~/.rbenv
```



```
vagrant@ubuntu-xenial:~$ cd
vagrant@ubuntu-xenial:~$ git clone https://github.com/rbenv/rbenv.gi
t ~/.rbenv
Cloning into '/home/vagrant/.rbenv'...
remote: Counting objects: 2714, done.
remote: Total 2714 (delta 0), reused 0 (delta 0), pack-reused 2714
Receiving objects: 100% (2714/2714), 506.25 KiB | 299.00 KiB/s, done
.
Resolving deltas: 100% (1700/1700), done.
Checking connectivity... done.
vagrant@ubuntu-xenial:~$
```

Ruby 개발환경 만들기

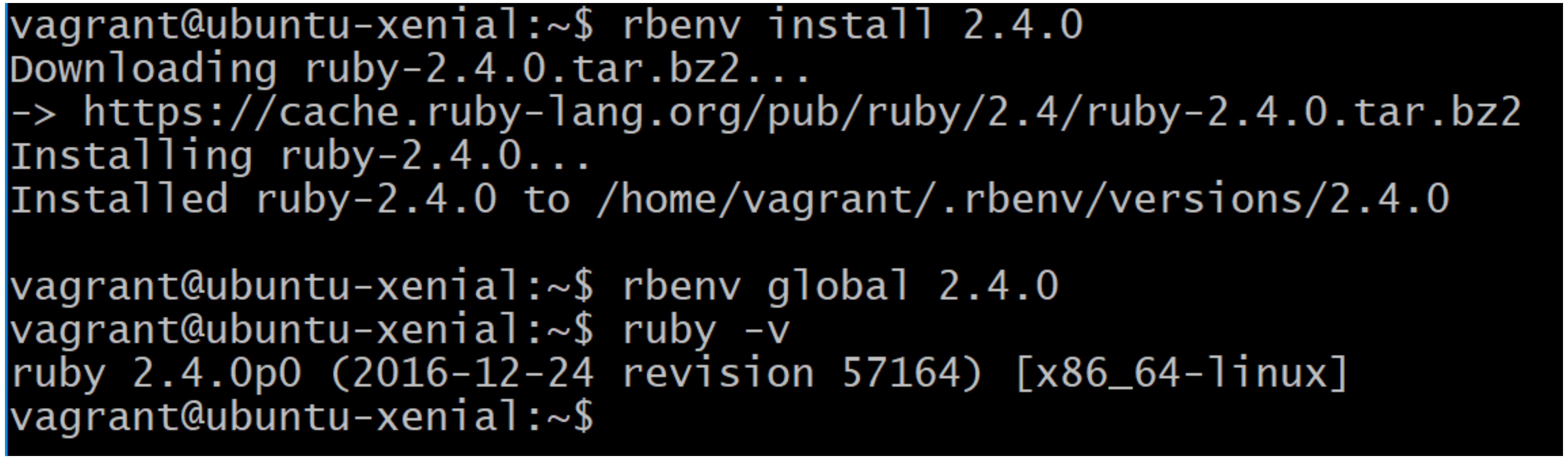
```
echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc
echo 'eval "$(rbenv init -)"' >> ~/.bashrc
exec $SHELL
git clone https://github.com/rbenv/ruby-build.git ~/.rbenv/plugins/ruby-build
echo 'export PATH="$HOME/.rbenv/plugins/ruby-build/bin:$PATH"' >> ~/.bashrc
exec $SHELL
```



```
vagrant@ubuntu-xenial:~$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"'
 >> ~/.bashrc
vagrant@ubuntu-xenial:~$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"'
 >> ~/.bashrc
vagrant@ubuntu-xenial:~$ echo 'eval "$(rbenv init -)"' >> ~/.bashrc
vagrant@ubuntu-xenial:~$ exec $SHELL
vagrant@ubuntu-xenial:~$ git clone https://github.com/rbenv/ruby-bui
ld.git ~/.rbenv/plugins/ruby-build
Cloning into '/home/vagrant/.rbenv/plugins/ruby-build'...
remote: Counting objects: 8557, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 8557 (delta 2), reused 0 (delta 0), pack-reused 8553
Receiving objects: 100% (8557/8557), 1.79 MiB | 804.00 KiB/s, done.
Resolving deltas: 100% (5452/5452), done.
Checking connectivity... done.
vagrant@ubuntu-xenial:~$ echo 'export PATH="$HOME/.rbenv/plugins/rub
y-build/bin:$PATH"' >> ~/.bashrc
vagrant@ubuntu-xenial:~$ exec $SHELL
vagrant@ubuntu-xenial:~$ |
```

Ruby 개발환경 만들기

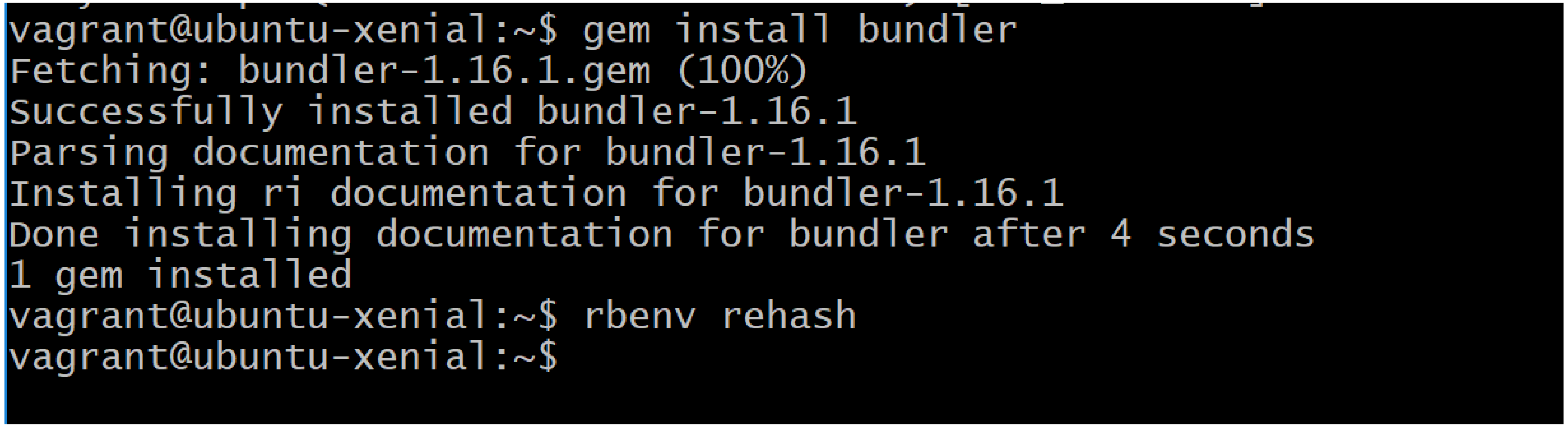
```
rbenv install 2.4.0
rbenv global 2.4.0
ruby -v
```



```
vagrant@ubuntu-xenial:~$ rbenv install 2.4.0
Downloading ruby-2.4.0.tar.bz2...
-> https://cache.ruby-lang.org/pub/ruby/2.4/ruby-2.4.0.tar.bz2
Installing ruby-2.4.0...
Installed ruby-2.4.0 to /home/vagrant/.rbenv/versions/2.4.0

vagrant@ubuntu-xenial:~$ rbenv global 2.4.0
vagrant@ubuntu-xenial:~$ ruby -v
ruby 2.4.0p0 (2016-12-24 revision 57164) [x86_64-linux]
vagrant@ubuntu-xenial:~$
```

Ruby 개발환경 만들기

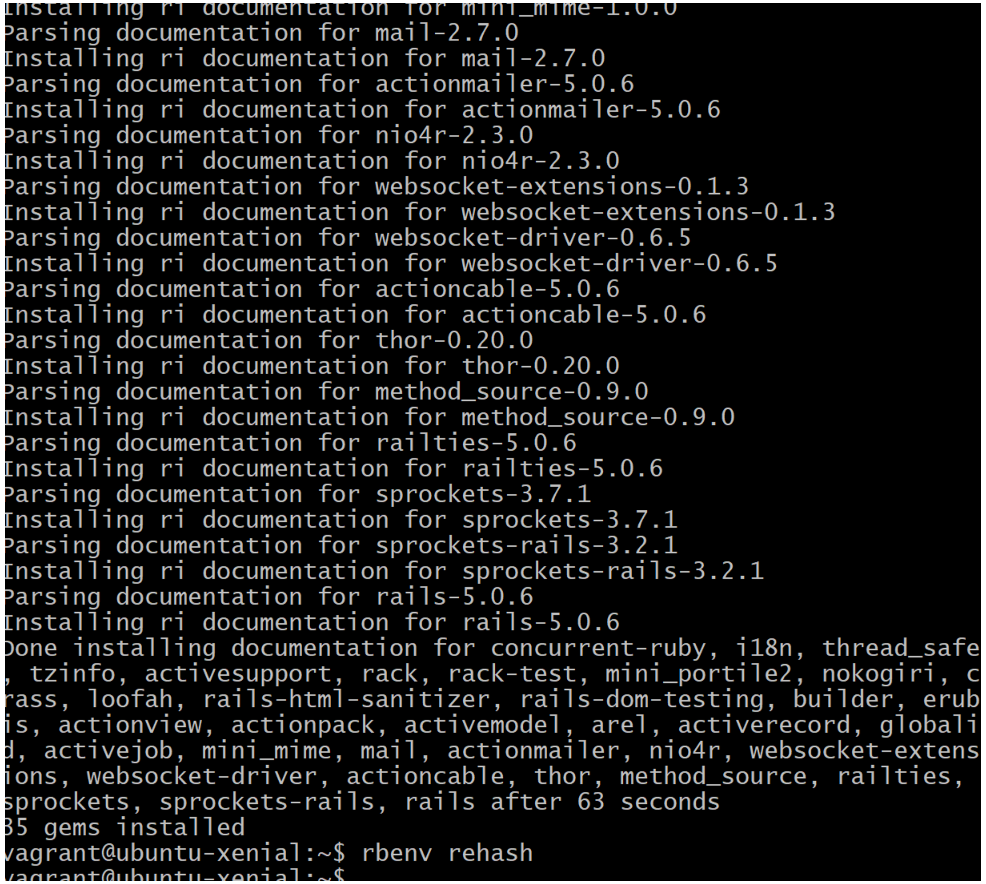
```
gem install bundler
rbenv rehash
```



```
vagrant@ubuntu-xenial:~$ gem install bundler
Fetching: bundler-1.16.1.gem (100%)
Successfully installed bundler-1.16.1
Parsing documentation for bundler-1.16.1
Installing ri documentation for bundler-1.16.1
Done installing documentation for bundler after 4 seconds
1 gem installed
vagrant@ubuntu-xenial:~$ rbenv rehash
vagrant@ubuntu-xenial:~$
```

Ruby 개발환경 만들기

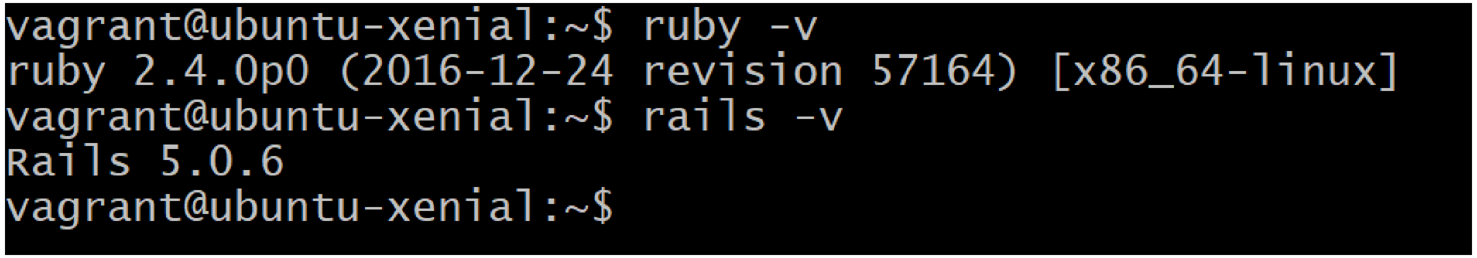
```
gem install rails -v 5.0.6
rbenv rehash
```



```
Installing ri documentation for mini_mime-1.0.0
Parsing documentation for mail-2.7.0
Installing ri documentation for mail-2.7.0
Parsing documentation for actionmailer-5.0.6
Installing ri documentation for actionmailer-5.0.6
Parsing documentation for nio4r-2.3.0
Installing ri documentation for nio4r-2.3.0
Parsing documentation for websocket-extensions-0.1.3
Installing ri documentation for websocket-extensions-0.1.3
Parsing documentation for websocket-driver-0.6.5
Installing ri documentation for websocket-driver-0.6.5
Parsing documentation for actioncable-5.0.6
Installing ri documentation for actioncable-5.0.6
Parsing documentation for thor-0.20.0
Installing ri documentation for thor-0.20.0
Parsing documentation for method_source-0.9.0
Installing ri documentation for method_source-0.9.0
Parsing documentation for railties-5.0.6
Installing ri documentation for railties-5.0.6
Parsing documentation for sprockets-3.7.1
Installing ri documentation for sprockets-3.7.1
Parsing documentation for sprockets-rails-3.2.1
Installing ri documentation for sprockets-rails-3.2.1
Parsing documentation for rails-5.0.6
Installing ri documentation for rails-5.0.6
Done installing documentation for concurrent-ruby, i18n, thread_safe
, tzinfo, activesupport, rack, rack-test, mini_portile2, nokogiri, c
rass, loofah, rails-html-sanitizer, rails-dom-testing, builder, erub
is, actionview, actionpack, activemodel, arel, activerecord, globali
d, activejob, mini_mime, mail, actionmailer, nio4r, websocket-extens
ions, websocket-driver, actioncable, thor, method_source, railties,
sprockets, sprockets-rails, rails after 63 seconds
35 gems installed
vagrant@ubuntu-xenial:~$ rbenv rehash
vagrant@ubuntu-xenial:~$
```

Ruby 개발환경 만들기

```
ruby -v
rails -v
```



```
vagrant@ubuntu-xenial:~$ ruby -v
ruby 2.4.0p0 (2016-12-24 revision 57164) [x86_64-linux]
vagrant@ubuntu-xenial:~$ rails -v
Rails 5.0.6
vagrant@ubuntu-xenial:~$
```

- Ruby와 rails의 버전을 확인하는 것으로 설치가 완료 되었음을 확인하였습니다.