

Vagrant BASE Box 만들기(CentOS 7)

2015.10.14. vagrant

Vagrant에는 [Discover Vagrant Boxes](#) 페이지를 제공하는데 여기에는 여러가지 Box가 이미 만들어져 있어 그중 하나를 선택해서 사용할 수 있다. 하지만 여기서 원하는 Box를 찾을 수 없거나 특별히 직접 만들어야 할 경우가 있다. 이 글은 이런 경우에 직접 Box를 만드는 방법에 대해 설명한다. 단, 아래는 OSX에서 작업하는 것을 기준으로 설명한다.

Vagrant 설치

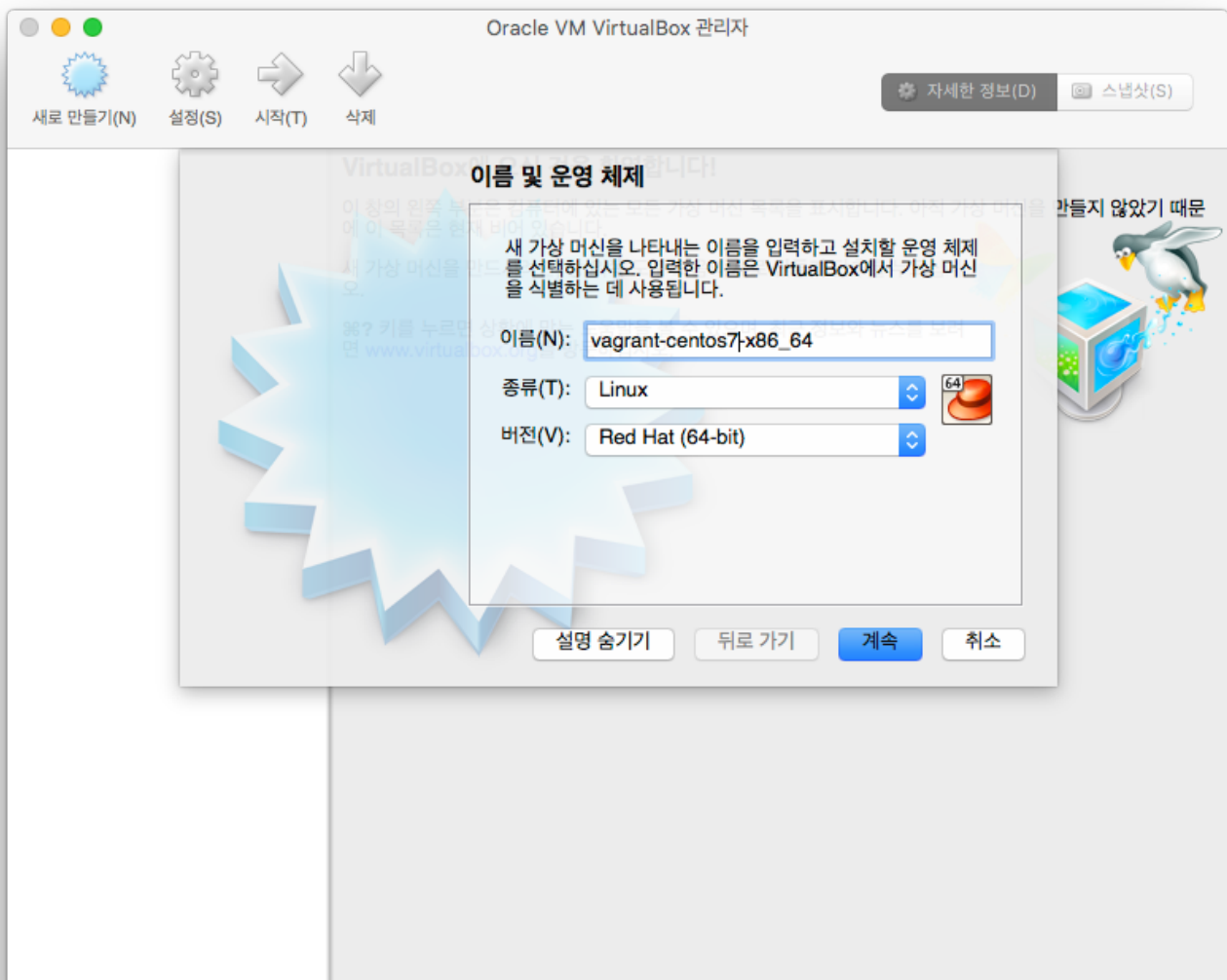
INSTALLING VAGRANT의 안내에 따라 [DOWNLOAD VAGRANT](#)에 가면 OS별로 Vagrant 설치 파일을 다운 받을 수 있다. 설치하는 다운 받아서 설치하면 되므로 굳이 설명할 것이 없다.

그리고 필요한 것이 [VirtualBox](#)인데 이것 또한 그냥 다운 받아 실행하면 설치가 완료되므로 간단하다.

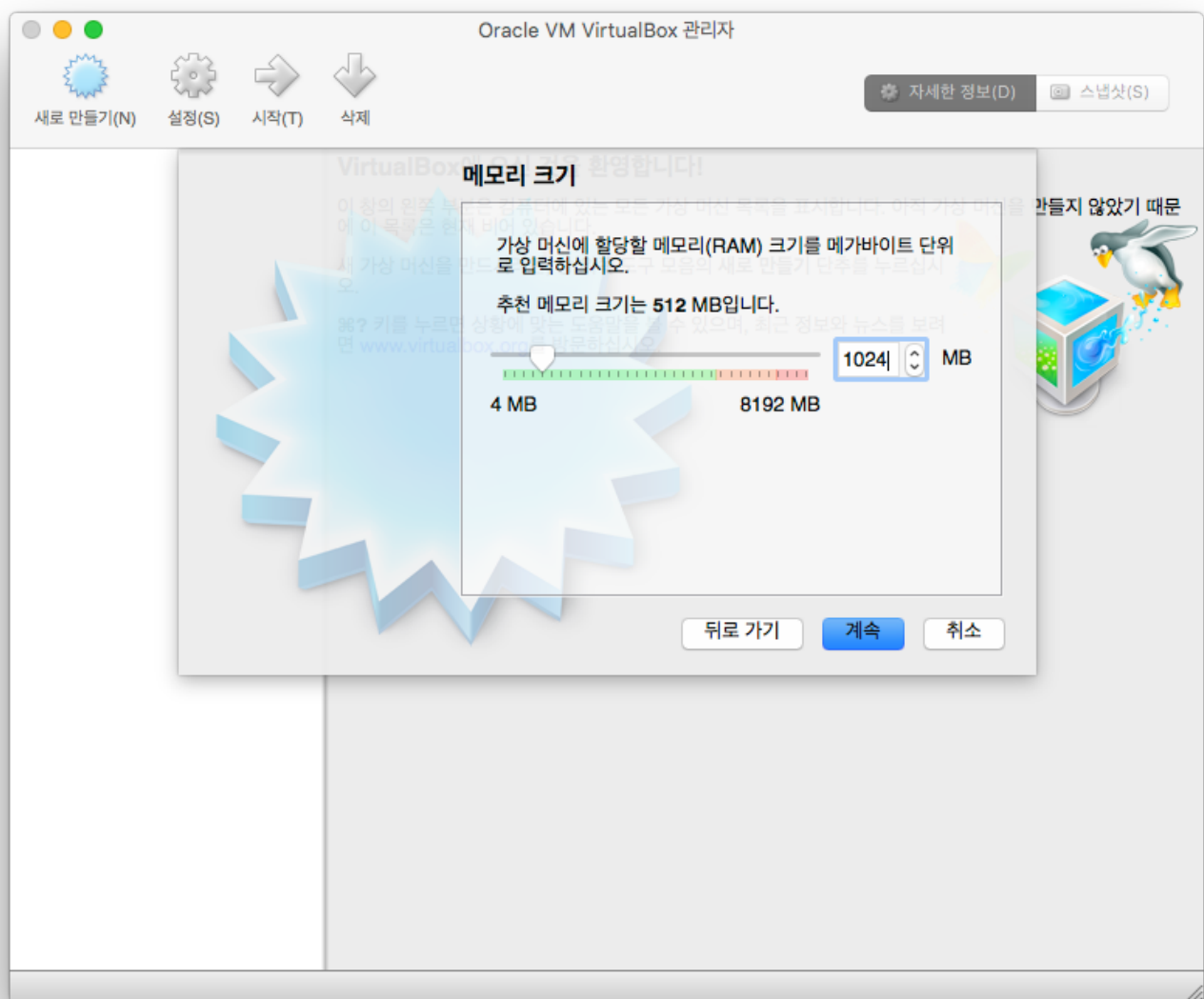
Base Box 만들기

VirtualBox 가상 머신 생성

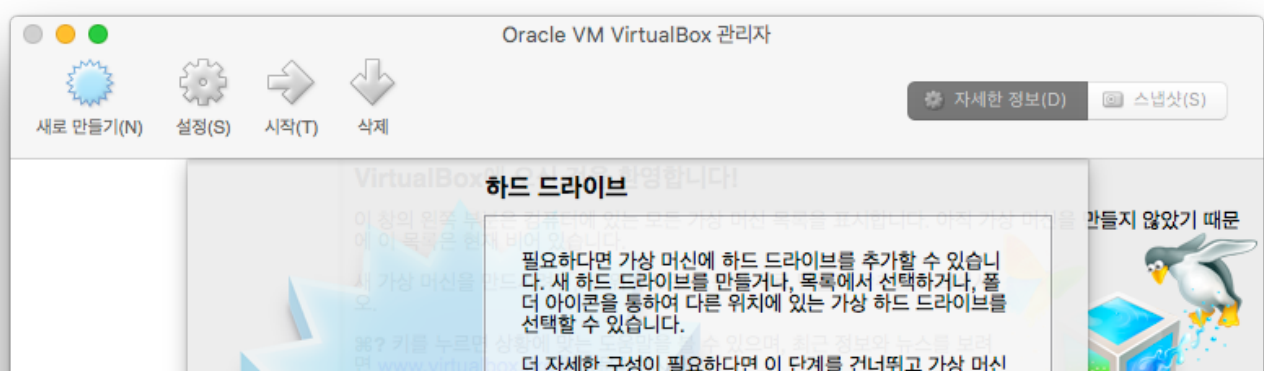
Box를 만드려면 VirtualBox 가상 머신이 필요하다. 우선 VirtualBox를 실행해서 “새로 만들기”를 선택하면 아래의 화면이 나타난다. 적절한 이름과 종류를 선택하고 계속을 선택한다. 여기서는 CentOS 7을 설치하는 것을 기준으로 설명한다.



그 다음엔 사용할 메모리 크기를 설정하는 화면이 나오는데 필요에 맞게 설정한다. 나는 1GB를 할당했다.

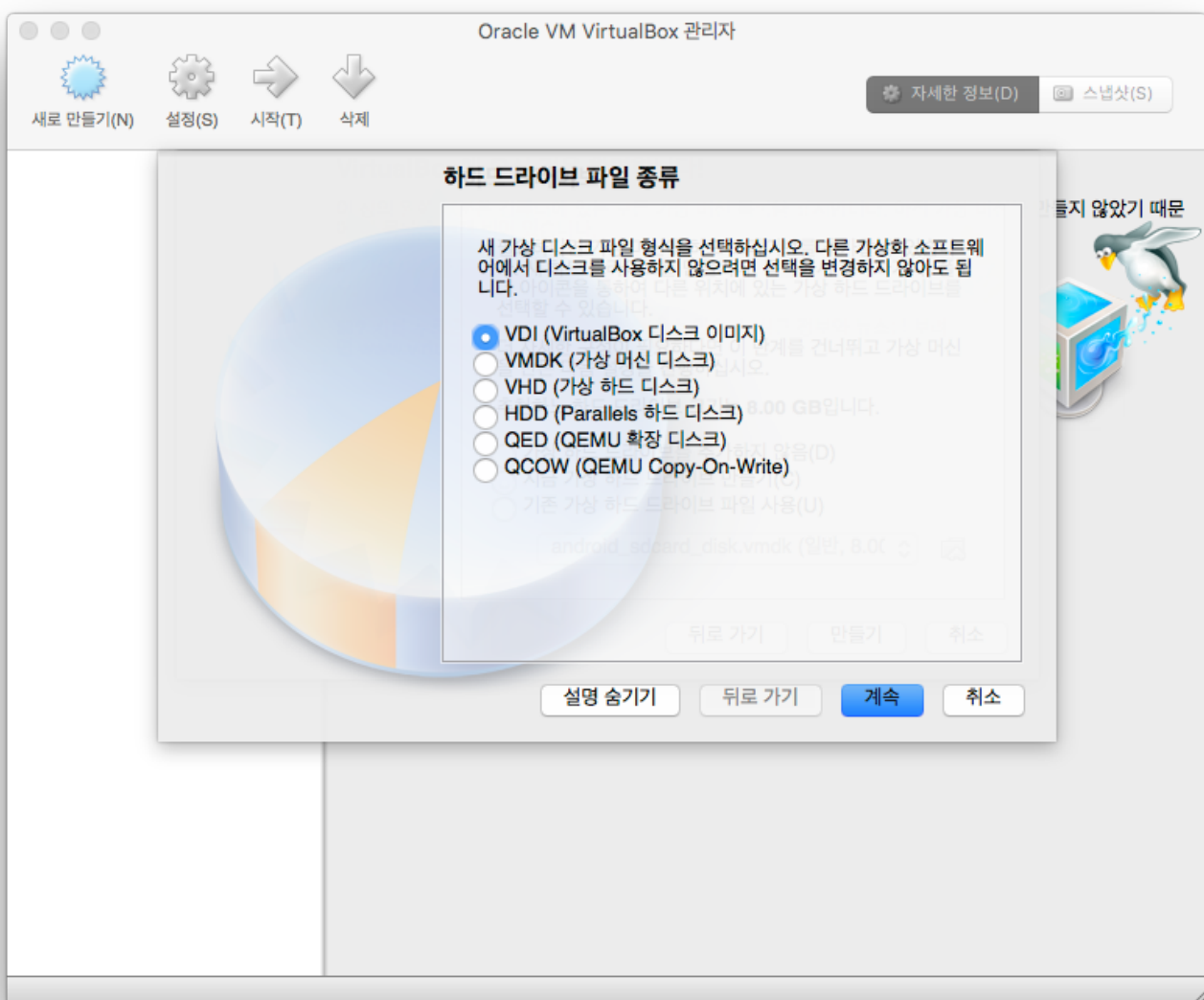


그 다음엔 가상 하드 드라이브를 설정하는 화면이다. 특별한 경우가 아니라면 “지금 가상 하드 드라이브 만들기”를 선택하면 된다.

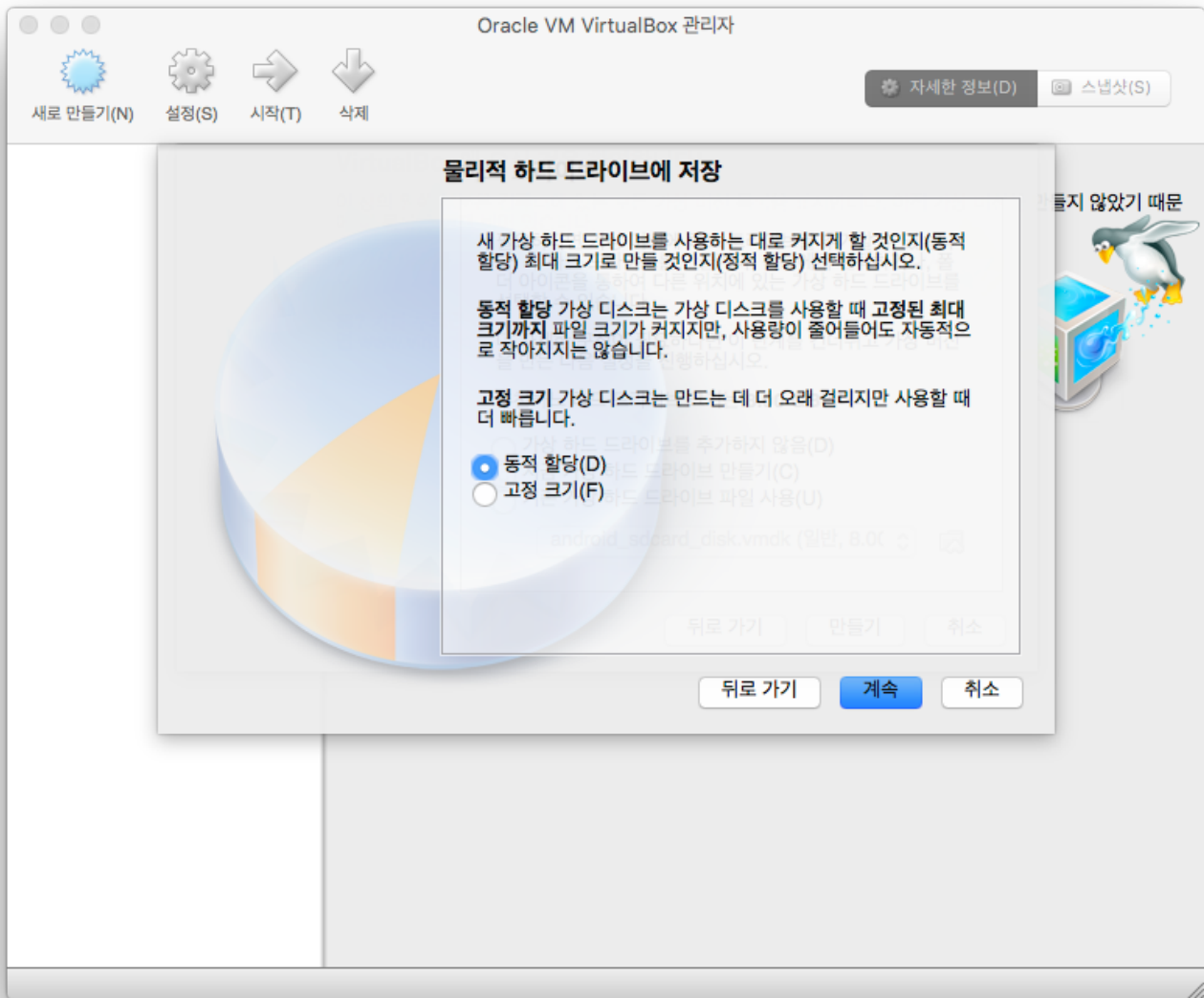




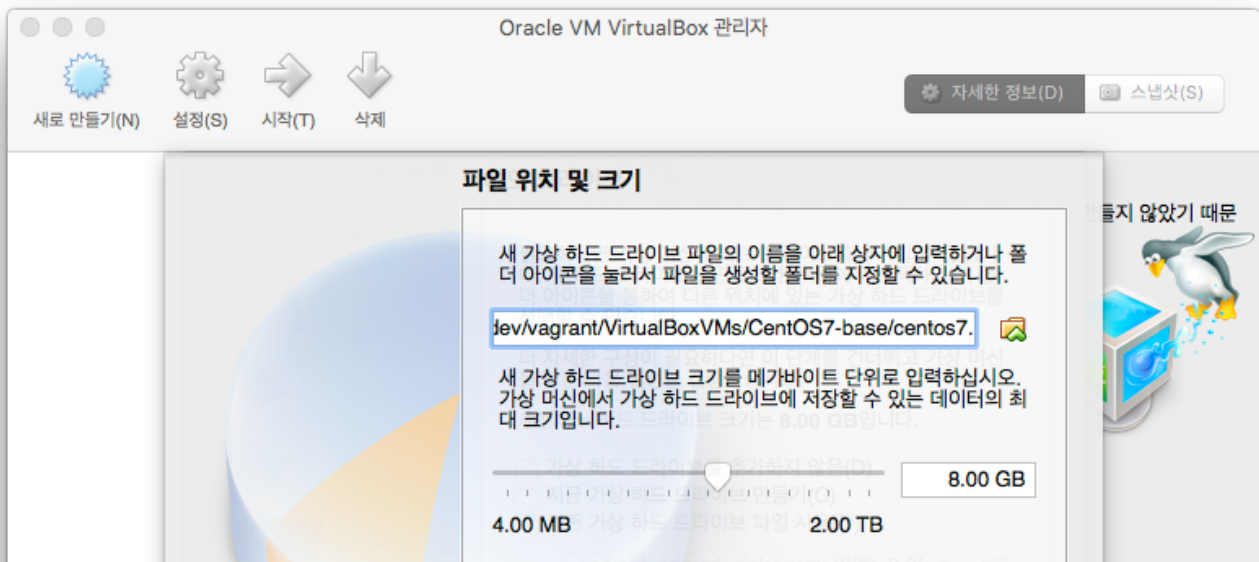
디스크 파일 형식의 경우는 기본으로 선택된 VDI를 선택한다.

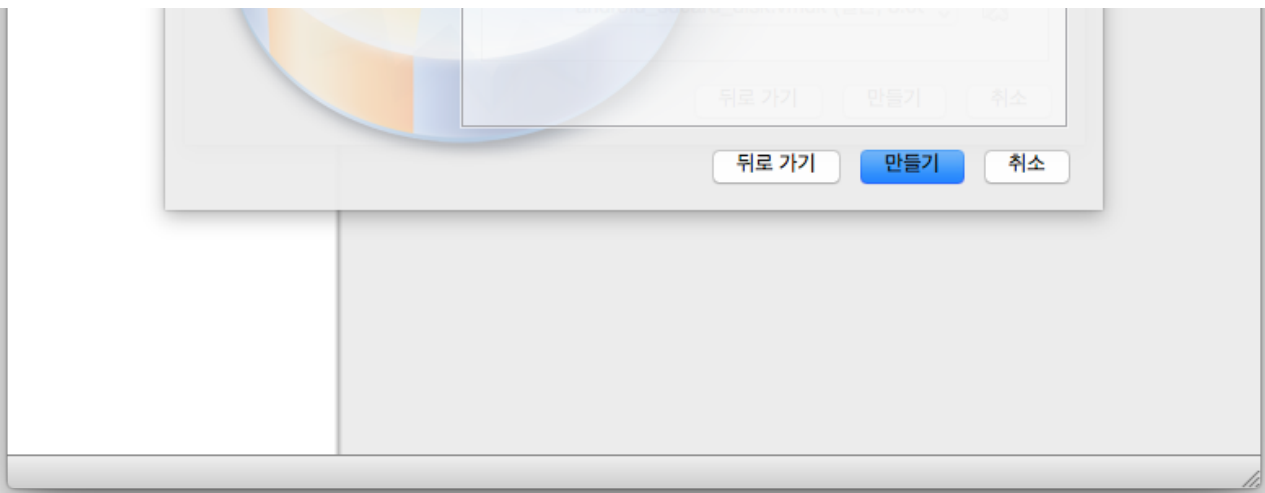


하드 드라이브의 크기는 특별한 이유가 없다면 동적 할당을 선택하면 된다.

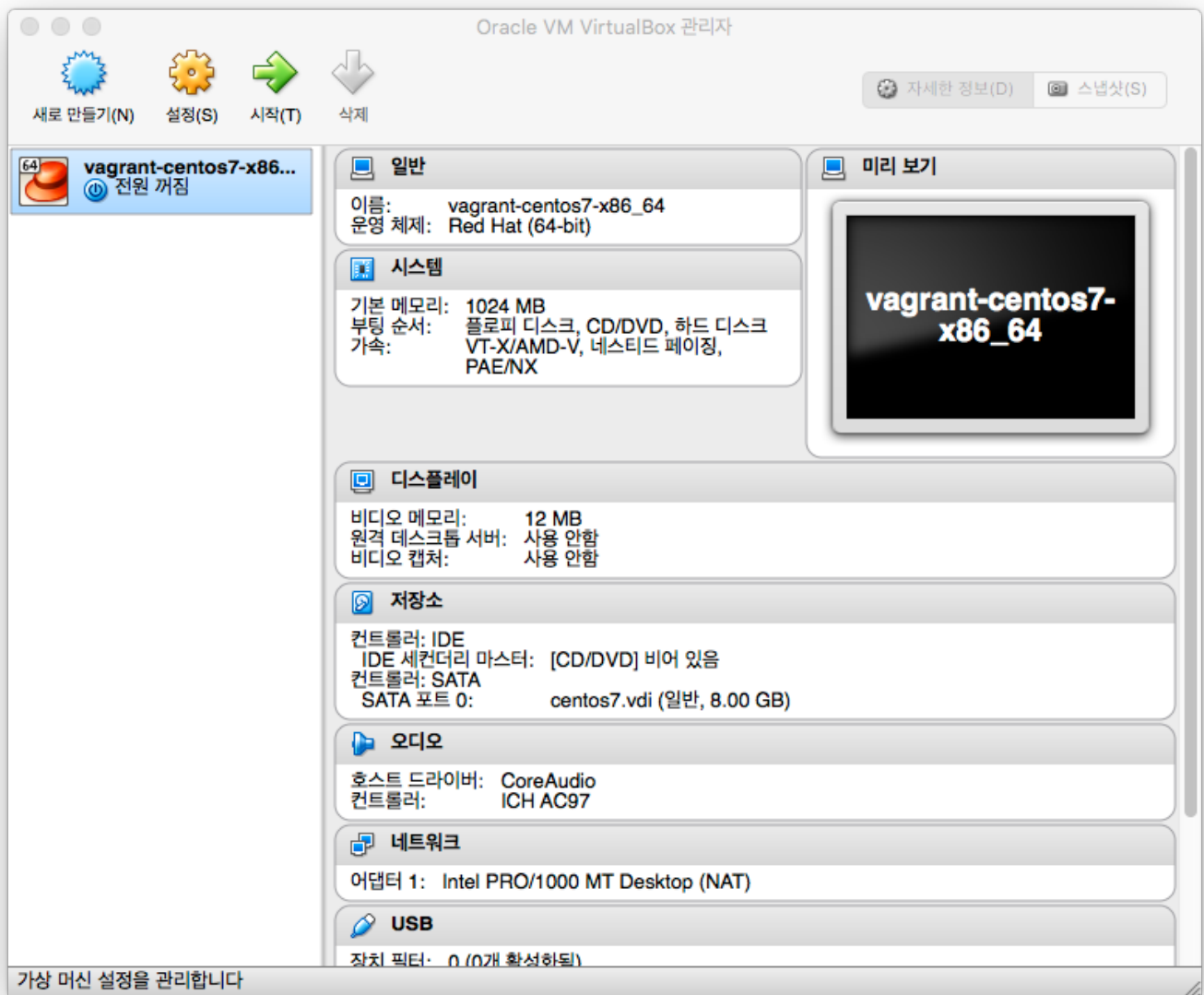


파일 위치 및 크기는 용도에 맞게 설정하면 된다.

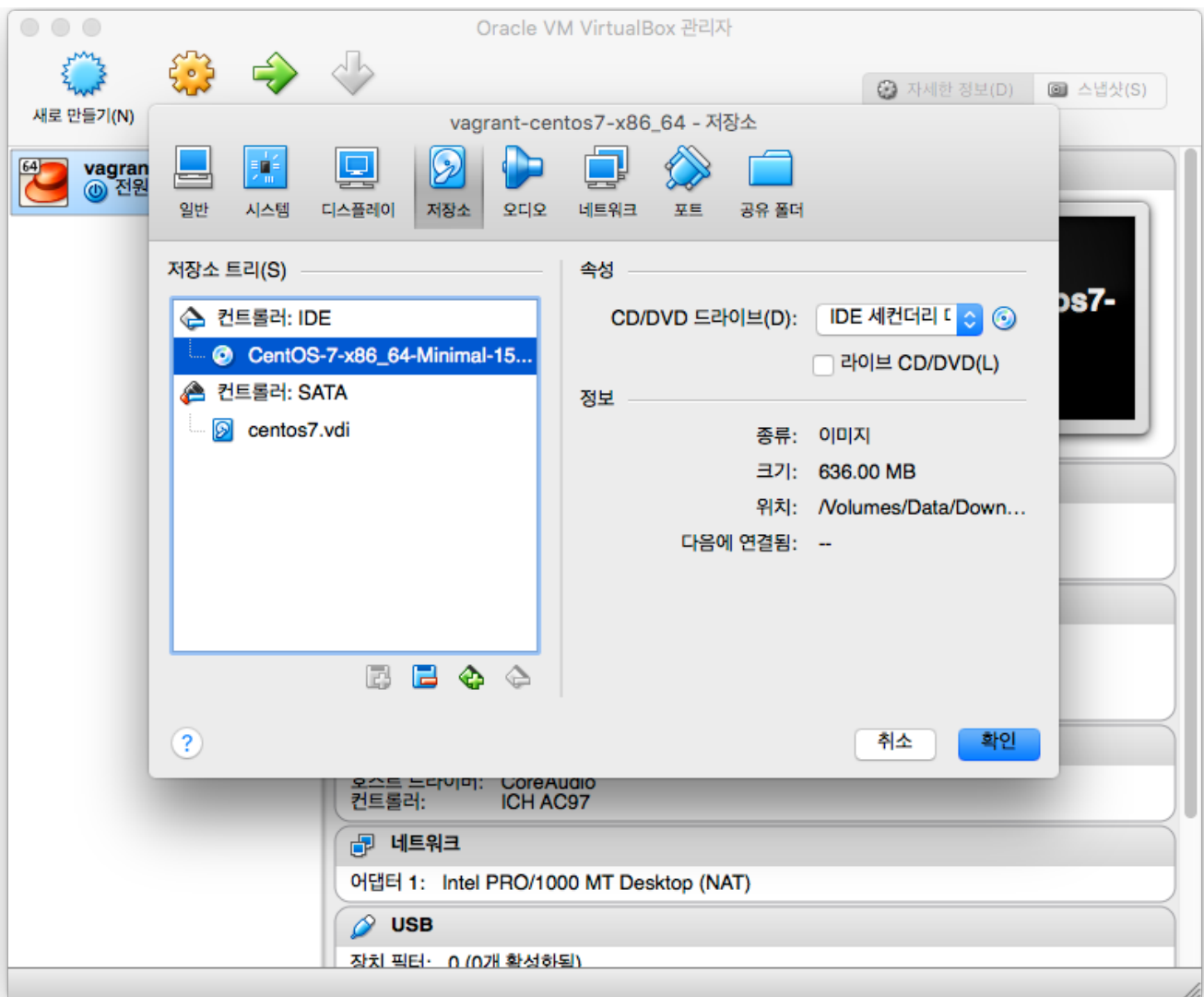




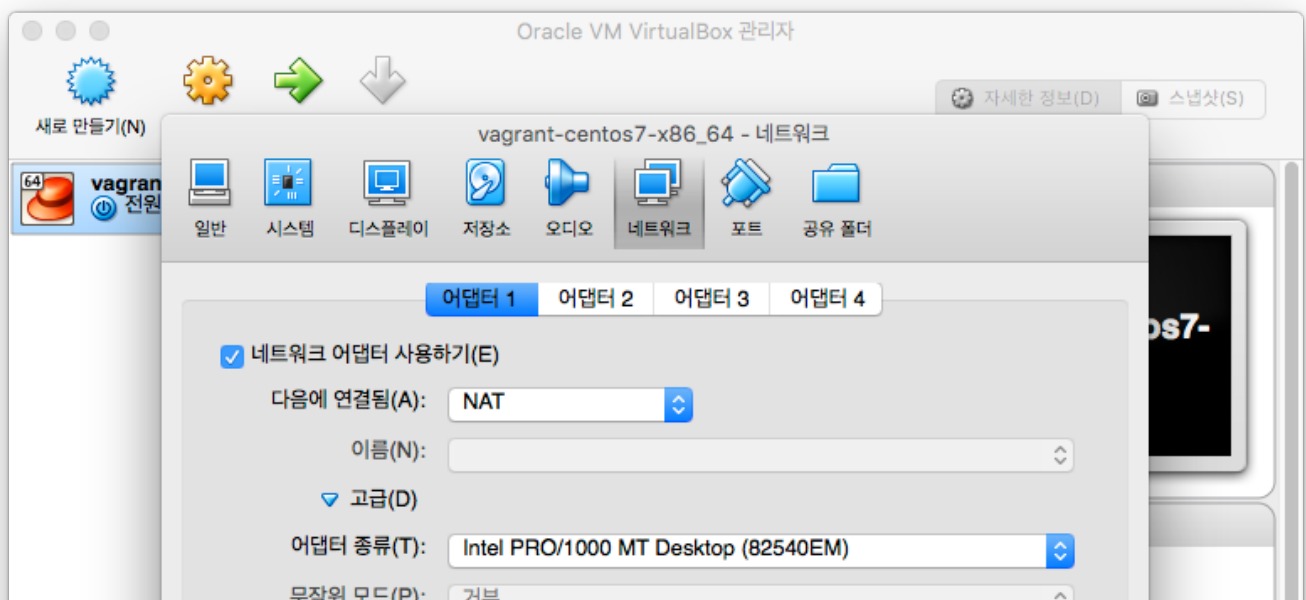
여기까지 설정하고 나면 가상 머신이 하나 만들어 진다.

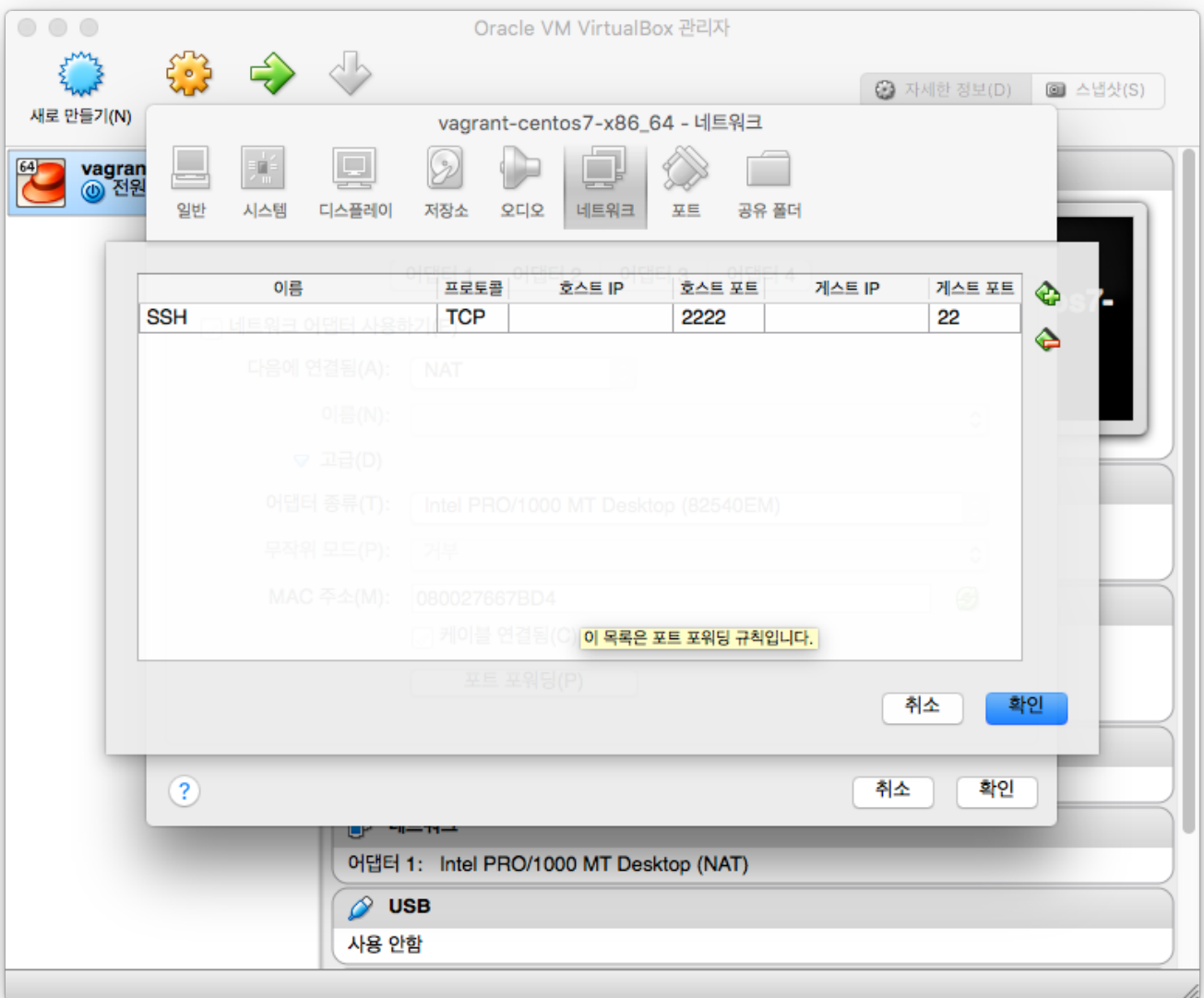
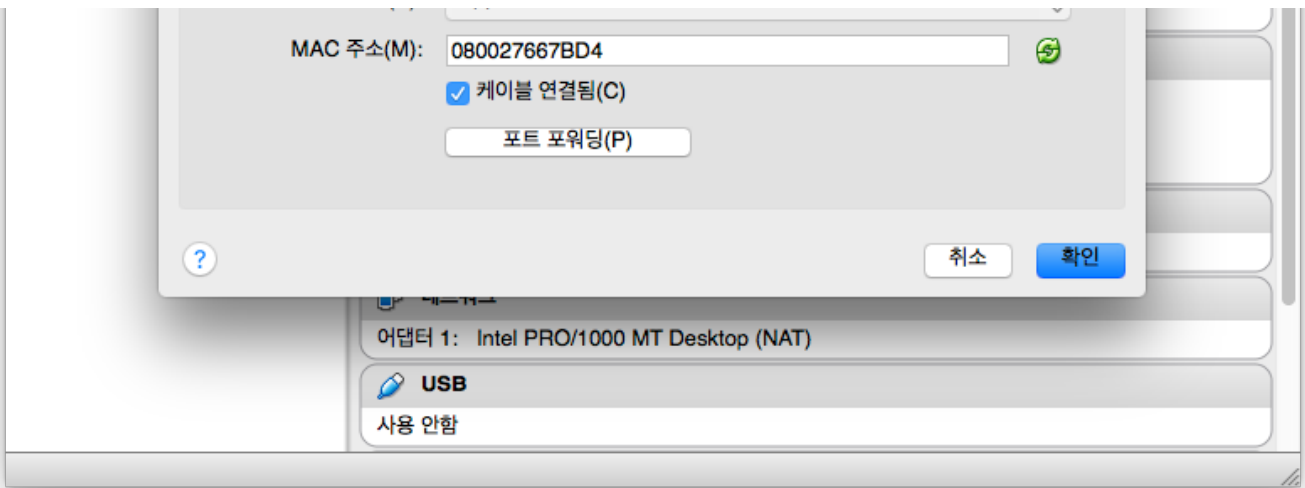


만들어진 가상 머신을 선택하고 설정을 선택하면 아래의 화면이 나온다.



여기서는 필요한 설정을 해주면 되는데 나의 경우엔 플로피디스크, USB, Audio를 사용하지 않도록 하고 네트워크는 “NAT”를 선택했다. “브릿지 어댑터”를 사용하면 내 컴퓨터의 IP를 함께 사용하지 않고 새로운 IP를 설정할 수 있는데 특수한 경우가 아니라면 일단 NAT를 사용하도록 설정한다. 그리고 NAT를 사용하게 되면 가상의 사설 IP가 할당되므로 나중에 SSH 접속을하기 위해서는 포트 포워딩 설정이 필요하다. 아래의 이미지를 참고해서 2222포트를 22포트로 포워딩 되도록 설정한다.





사용할 OS 설치

이제는 설치할 OS를 지정해 줘야 한다. 설정 화면에서 저장소를 선택하면 컨트롤러 IDE에 CD 드라이브가 보일 것이다. 나는 CentOS 7을 설치할 것이므로 해당 ISO 파일을 다운 받아서 선택해 줬다.

이제 설정을 완료하고 서버를 시작하면 설정 해준 ISO(CD)가 로딩되면서 OS 설치가 시작될 것이다. 내

가 설치한 CentOS 7의 설치 과정은 **CentOS 7 minimal 설치**를 참고하면 된다.

OS에 대한 기본 설치를 마쳤다면 VirtualBox에서 해당 가상 머신을 켜 상태에서 메뉴의 “Devices > Insert Guest Additions CD Image...”를 선택한다. 그 다음엔 콘솔에서 아래의 명령을 수행해서 게스트 확장을 설치한다. 그리고 설치가 정상적으로 완료되었다면 재시작 한다(굳이 할 필요는 없지만 kernel과 관련된 변경이 있는 것으로 보여 재시작 해 줬다).

```
$ su
$ mount -r /dev/cdrom /media/
$ ll /media/
$ yum install -y bzip2 kernel-devel make gcc perl
$ /media/VBoxLinuxAdditions.run --nox11
$ yum clean all

$ usermod -u 471 vboxadd
$ groupmod -g 471 vboxsf
#lsmod | grep vboxsf

$ export MAKE='/usr/bin/gmake -i'
$ /etc/init.d/vboxadd setup
$ unset MAKE

$ reboot
```

간혹 VBoxLinuxAdditions.run을 실행하는 과정에 kernel-devel이 없다는 오류가 날 수 있다. 이 경우엔 오류 메시지에 kernel-devel의 특정 버전을 yum으로 설치하라는 내용이 포함되어 있다. 이 명령을 따라 kernel-devel을 설치하고 VBoxLinuxAdditions.run를 다시 실행하면 오류없이 설치가 된다.

Vagrant 관련 설정

우선 vagrant 계정을 추가한다.

```
$ /usr/sbin/groupadd -g 470 vagrant
$ /usr/sbin/useradd -g vagrant -u 470 vagrant
$ passwd vagrant
# vagrant 를 암호로 입력
$ chown vagrant.vagrant /home/vagrant/
```

vagrant 계정에서 root 명령을 수행할 수 있도록 sudo에 추가해 준다.

```
$ visudo
# 아래 내용 추가
Defaults    env_keep += "SSH_AUTH_SOCK"
%vagrant    ALL=(ALL)        NOPASSWD: ALL
Defaults:vagrant !requiretty
```

sudo에 정상적으로 추가되었는지를 확인하기 위해 아래의 명령을 수행해 본다.

```
$ su - vagrant
```



```
$ su vagrant
$ cd
$ sudo ls -al
```

vagrant의 ssh key를 추가해 준다. 위의 확인 코드를 수행했다면 /home/vagrant에 위치하고 있을 것이다. 아니라면 해당 폴더로 이동한다.

```
$ mkdir .ssh
$ curl -k https://raw.githubusercontent.com/mitchellh/vagrant/master/keys/vagrant.pub > .ssh/authorized_keys
$ chmod 0700 .ssh
$ chmod 0600 .ssh/authorized_keys
# .ssh/authorized_keys 내용 확인
$ cat .ssh/authorized_keys
```

여기서 주의할 것이 있다. authorized_keys 파일이 빈 파일로 저장되는 경우가 있다. 정확한 원인은 확인하지 못했지만 이번에 작업하다보니 이런 경우가 생겼다. 현재로서는 curl의 문제로 보이지만 확실하진 않다. 이 경우엔 일단 파일을 받아야 하니 아래와 같이 wget을 사용해서 받자.

```
$ sudo yum install wget -y
$ mkdir .ssh
$ wget --no-check-certificate https://raw.githubusercontent.com/mitchellh/vagrant/master/keys/vagrant.pub > .ssh/authorized_keys
$ chmod 0700 .ssh
$ chmod 0600 .ssh/authorized_keys
# .ssh/authorized_keys 내용 확인
$ cat .ssh/authorized_keys
```

이제는 ssh 설정을 조정한다.

```
$ sudo vi /etc/ssh/sshd_config
# 아래의 내용을 확인해서 맞춰준다.
Port 22
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
```

디스크를 정리해서 용량을 줄인다. 디스크 용량이 부족하다는 오류가 나올 수 있는데 무시하면 된다. 디스크가 가득 찼때까지 내용을 채우기 때문이라고 한다.(<http://stackoverflow.com/a/24956805>) 이렇게 하면 가상이미지를 Box로 변환시 용량을 줄일 수 있다고 한다.

```
sudo dd if=/dev/zero of=/EMPTY bs=1M
sudo rm -f /EMPTY
```

이제 기본 설정을 마쳤으므로 서버를 종료한다.

Box 만들기

이제 마지막 단계다. OSX의 shell에서 아래 명령을 입력하면 설치된 VirtualBox 목록이 출력된다. 여기서 생성했던 가상 머신의 이름을 확인한다.

이제 박스를 만든다. “vagrant-centos7-x86_64”는 위에서 확인한 VirtualBox 이름을 넣으면 되고

“centos7-x86_64.box”는 생성할 박스의 이름을 넣으면 된다.

```
vagrant package --output centos7-x86_64.box --base vagrant-centos7-x86_64
```

혹시 퍼미션 오류가 난다면 아래처럼 sudo를 사용하면 된다. 사실 일반적인 경우라면 당연히 퍼미션 오류가 나지 않는다. 그런데 나의 경우엔 1024 이하의 포트를 포트 포워딩하기 위해 vagrant와 VirtualBox를 root 권한에서 사용하고 있다(OSX에서는 1024 포트를 포트 포워딩하기 위해서는 root 권한이 필요하다). 이런 경우라면 vagrant 명령에서 권한 오류가 날 수 있다.

```
sudo vagrant package --output centos7-x86_64.box --base vagrant-centos7-x86_64
```

Vagrantfile 만들기

아래의 명령을 실행하면 Vagrantfile가 만들어 진다. 아래는 두가지 방법을 설명하고 있다. 첫번째는 local에 box를 추가해서 init하는 것이고 두번째는 현재 폴더의 box 파일을 사용해서 init하는 것이다. 어떤 방법을 써도 Vagrantfile는 만들어지니 편한 방법을 사용하면 된다. 어쨌든 vagrant up 실행시 박스는 만들어진다. 보통 “~/.vagrant.d/boxes”에 Box 파일이 추가된다.

```
$ vagrant box add centos7 centos7-x86_64.box
$ vagrant init centos7
```

또는

```
$ vagrant init centos7-x86_64.box
# or
$ vagrant init
```

여기서 만들어진 파일은 가장 기본 설정만 되어 있으므로 자신에게 맞는 설정을 해주어야 한다. 이 부분은 또 내용이 길기 때문에 다음에 포스팅 하도록 하겠다.

실행해보기

앞선 과정에서 문제가 없었다면 정상적으로 vagrant가 실행된다. 당연히 최소 실행시에는 box를 다운 받는 과정이 있어 실행이 좀 오래 걸린다.

아래의 명령으로 접속이 된다면 일단 성공이다.

박스의 종료.


추가된 box를 제거하려면 아래를 참고하자.

```
$ vagrant destroy
$ vagrant box list
$ vagrant box remove centos7-x86_64.box
```

destroy는 VirtualBox에서 가상 머신을 제거한다. 하지만 추가된 box는 제거되지 않는다. 따라서 box list를 이용해서 box명을 확인해서 box remove를 해주어야 box까지 제거된다. 여기서 제거란 설치된 파일이 제거를 뜻하므로 마더어두 box파일과 Vagrantfile 파일은 그대로 유지된다

 **blog.asamaru.net**

 <https://blog.asamaru.net/2015/10/14/creating-a-vagrant-base-box/>

 <http://goo.gl/ffgS>

