

VirtualBox 및 Vagrant 설치

1. VirtualBox 설치(virtualbox.org)

* hyper-v 가 실행중이면 hyper-v를 종료해야 한다.

- About
- Screenshots
- Downloads
- Documentation
 - End-user docs
 - Technical docs
- Contribute
- Community

Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.1 packages, see [VirtualBox 6.1 builds](#). Version 6.1 will remain supported until December 2023.

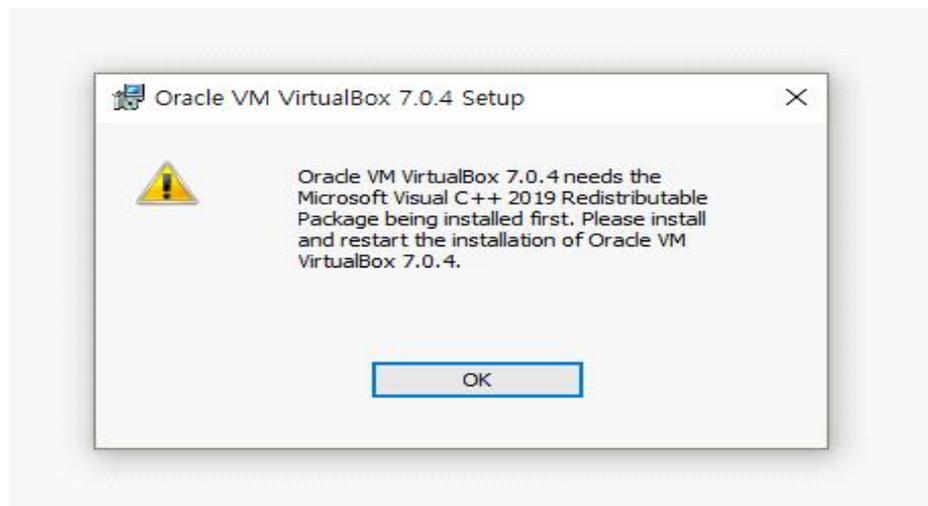
VirtualBox 7.0.10 platform packages

- [Windows hosts](#)
- [macOS / Intel hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)
- [Solaris 11 IPS hosts](#)

The binaries are released under the terms of the GPL version 3.

See the [changelog](#) for what has changed.

virtualbox 설치시 아래와 같은 에러메시지가 출력이되면



Microsoft Visual C++ 2019를 설치하면 된다.

<https://learn.microsoft.com/ko-kr/cpp/windows/latest-supported-vc-redist?view=msvc-170>

아키텍처	링크	참고
ARM64	https://aka.ms/vs/17/release/vc_redist.arm64.exe	지원되는 최신 ARM64 버전에 대한 Permalink
X86	https://aka.ms/vs/17/release/vc_redist.x86.exe	지원되는 최신 x86 버전에 대한 Permalink
X64	https://aka.ms/vs/17/release/vc_redist.x64.exe	지원되는 최신 x64 버전에 대한 Permalink입니다. X64 재배포 가능 패키지는 ARM64 및 X64 이진 파일이 모두 포함되어 있습니다. 이 패키지를 사용하면 X64 재배포 가능 패키지가 ARM64 디바이스에 설치될 때 필요한 Visual C++ ARM64 이진 파일을 쉽게 설치할 수 있습니다.

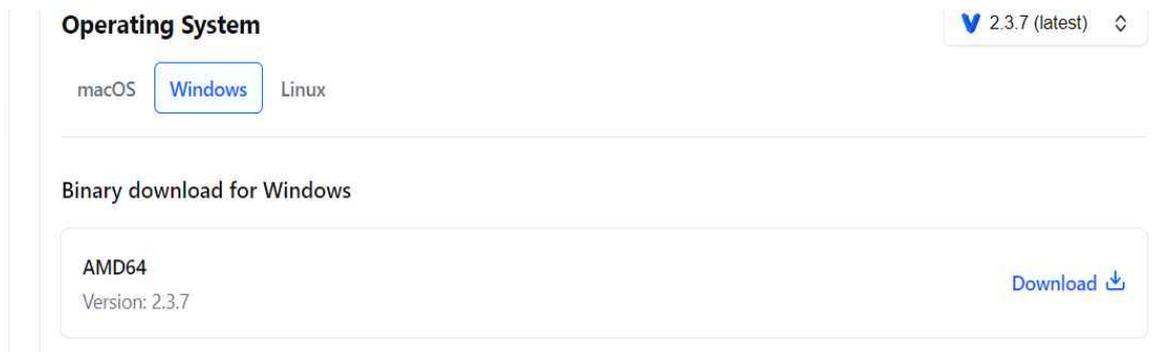
=>

*. virtualbox를 제대로 사용하기 위해서는 바이오스 설정에서 cpu 가상화 기능이 켜져 있어야 한다.

vagrant install(mswindows 기준)

<https://developer.hashicorp.com/vagrant/downloads>

메뉴중 Binary download for windows => AMD64 => Download



Download 받은 Vagrant를 설치하고 설치완료되면 리부팅 하라는 메시지가 출력됩니다. 반드시 리부팅해서 사용해야 합니다.

Vagrant program은 command line에서 실행하는 프로그램입니다. 파워셸이나 cmd에서 실행을 해야 합니다.

vagrant 명령어

vagrant image login id 및 암호.

id: **vagrant**

password: **vagrant**

* vagrant 계정이 암호인증 없는 sudo 권한을 갖고 있습니다.

vagrant 기본명령어

- vagrant 명령은 cmd 나 powershell 에서 사용합니다.

vagrant 설정파일명은 Vagrantfile 입니다.

vagrant 명령은 vagrantfile 이 있는 디렉토리에서 실행을 해야 합니다.

* 현재 디렉토리에 Vagrantfile 이 없으면 부모디렉토리를 참조합니다.

vagrant --help ; 도움말 출력

vagrant **init** => vagrant 실행을 위한 Vagrantfile 생성

vagrant **up** : Vagrantfile 내용을 읽고 실행

vagrant **status** : vagrant 가상머신 상태 확인

vagrant **halt** : vagrant 가상머신 종료

vagrant **destory** : 설치된 vagrant 가상머신 삭제

vagrant **provision** : Vagrantfile 에서 provision 항목 실행

vagrant **ssh** : vagrant 가상머신으로 ssh 접속

* 설치된 Vagrant vm image 삭제

Vagrantfile 이 있는 디렉토리안에서 아래의 명령어를 차례대로 실행

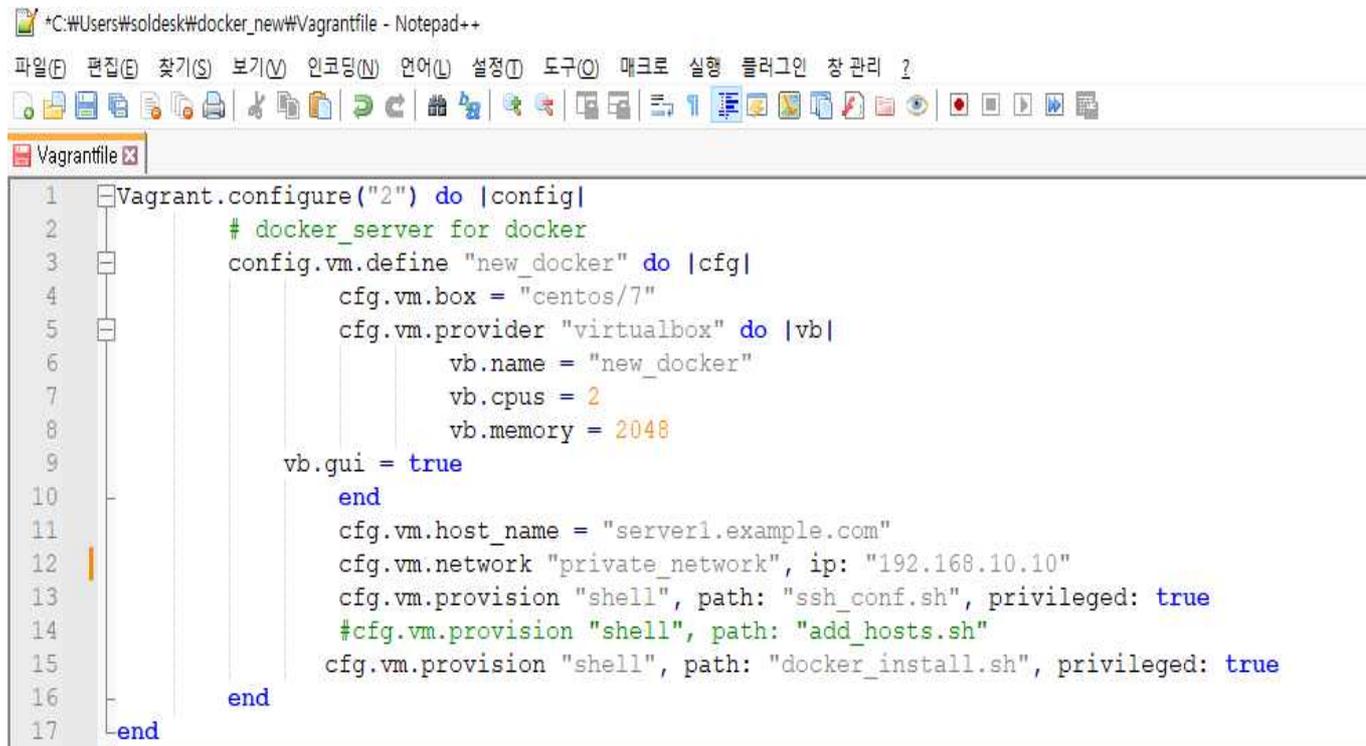
vagrant halt ; vagrant 가상머신 종료

vagrant status ; vagrant vm 상태 확인

vagrant destroy ; vagrant vm image 삭제

rm .vagrant ; vm 생성될때 만들어진 .vagrant 디렉토리 삭제

vagrant samplefile 참고 => <https://developer.hashicorp.com/vagrant/docs/vagrantfile>



```
1 Vagrant.configure("2") do |config|
2   # docker_server for docker
3   config.vm.define "new_docker" do |cfg|
4     cfg.vm.box = "centos/7"
5     cfg.vm.provider "virtualbox" do |vb|
6       vb.name = "new_docker"
7       vb.cpus = 2
8       vb.memory = 2048
9     end
10    vb.gui = true
11  end
12  cfg.vm.host_name = "server1.example.com"
13  cfg.vm.network "private_network", ip: "192.168.10.10"
14  cfg.vm.provision "shell", path: "ssh_conf.sh", privileged: true
15  #cfg.vm.provision "shell", path: "add_hosts.sh"
16  cfg.vm.provision "shell", path: "docker_install.sh", privileged: true
17 end
```

Vagrant.configure("2") => 숫자2는 버전.

config.vm.define => vagrant 가 관리할 이름이 된다. 중복되면 설치가 되지 않는다.
시스템 전체에서 유일해야 한다.

cfg.vm.box => vagrant image 운영체제타입

vb.name => virtualbox에서 보여지는 가상머신 이름.

vb.cpus => cpu 개수(또는 cpu코어갯수) 설정. vb.memory => 가상머신 메모리 크기 설정

cfg.vm.network "private_network" => virtualbox에서 hostonly 타입으로 설정됨.

ip address 는 충돌나지 않는 대역으로 설정하면 됨.

public_network 로 설정하면 virtualbox에서 bridge type 으로 설정된다.

cfg.vm.provision => vagrant image 가 실행이 될 때 실행할 명령어나 shell 스크립트를 적으면 실행이 됨.

privileged: true => true 가 디폴트값이므로 생략해도 된다. true 이면 명령어나 셸스크립트를 root 권한으로 실행
false 이면 vagrant 계정으로 실행이 됨.

cfg.vm.host_name => 리눅스 hostname 이 된다.

do ~ end 는 블록시작('{') 과 블록끝을('}') 의미한다