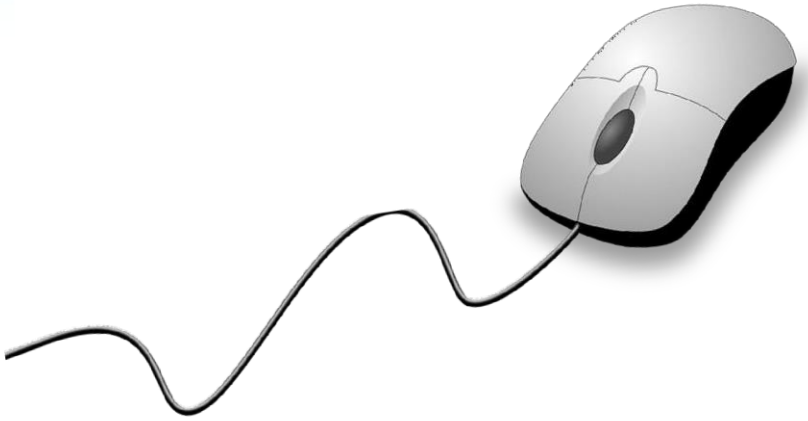
공개SW 솔루션 설치 & 활용 가이드

미들웨어 > 클라우드 서비스



제대로 배워보자

How to Use Open Source Software

Open Source Software Installation & Application Guide

오픈소스 소프트웨어 통합지원센터
Open Source Software Support Center

CONTENTS

1. 개요

소개	• Docker(도커)는 2013년 3월 Docker, Inc에서 출시한 오픈 소스 컨테이너 프로젝트 • 현재 전세계적으로 큰 인기 끌고 있으며 AWS, Google Cloud Platform, Microsoft Azure 등의 클라우드 서비스에서 공식 지원		
주요기능	• Docker 이미지 생성, 컨테이너 동작, Docker 이미지 공개 및 공유		
대분류	• 미들웨어	**소분류**	• 클라우드 서비스
라이선스 형태	• 아파치 라이선스 2.0	**사전설치 솔루션**	• N/A
실행 운영체제	• Windows, Linux	**버전**	• 18.05.0-ce / 2018년 5월 9일
특징	• Docker는 반가상화보다 경량화 방식이며 이미지 용량 작음 • 게스트 OS가없고Docker 이미지에 Application과 Library만 격리해서OS자원 호스트와공유 • 하드웨어 가상화 계층이 없기 때문에 메모리 접근, 파일시스템이 가상머신에비해 월등히빠름 • 가상 머신과 다르게 이미지 생성/배포에 특화된 이미지 버전 관리 기능 제공 • 중앙 관리 위해 저장소에 이미지 올리고, 받 수 있음(Push/Pull) • GitHub처럼 Docker 이미지 공유할 수 있는 Docker Hub 제공		
보안취약점	• 취약점 ID : CVE-2014-5282 • 심각도 : 8.1 HIGH(V3) • 취약점 설명 : 1.3 이전의 Docker는 이미지 ID의 유효성 제대로 검사하지 못하기 때문에 원격 공격자가 '도커로드' 통해 신뢰할 수없는 이미지 로드하여 다른 이미지로 리디렉션 할 수 있음 • 대응방안 : Docker 엔진 1.3으로 업데이트 • 참고 경로 : https://bugzilla.redhat.com/show_bug.cgi?id=1168436		
개발회사/커뮤니티	• https://www.docker.com/ / https://www.docker.com/docker-community		
공식 홈페이지	• https://www.docker.com		

2. 기능요약

- Docker 의 주요 기능

운영 표준화	• 작은 컨테이너식 애플리케이션 사용하면 손쉽게 배포하고, 문제 파악하고, 수정 위해 롤백 할 수 있음
지속적인 통합 및 제공	• 환경 표준화하고 언어 스택 및 버전 간의 충돌 제거함으로써 애플리케이션 더욱 빠르게제공
마이크로 서비스	• Docker 컨테이너 통해 표준화된 코드 배포 활용하여 분산 애플리케이션 아키텍처 구축하고 확장

3. 실행환경

- 설치 가능한 운영체제

구분	버전	하드웨어 아키텍쳐
Windows	Windows 10 64bit: Pro, Enterprise or Education (1607 Anniversary Update, Build 14393 or later)	x86_64
MacOS	El Capitan 10.11 이상	2010년도 모델 또는 신모델
Fedora	26, 27, 28	x86_64(또는 amd64)
CentOS	7	x86_64(또는 amd64)
Debian	Buster 10 (Docker CE 17.11 Edge only) Stretch 9 (stable) / Raspbian Stretch Jessie 8 (LTS) / Raspbian Jessie Wheezy 7.7 (LTS)	x86_64(또는 amd64), armhf
Ubuntu	Bionic 18.04 (LTS) Xenial 16.04 (LTS) Trusty 14.04 (LTS)	x86_64, armhf, s390x (IBM Z), ppc64le (IBM Power)

4. 설치 및 실행

세부 목차

4. 설치 및 실행

4.1 설치 진행(CentOS 설치1/4)

- yum 사용하여 패키지 설치
  - $ sudo yum install -y yum-utils ₩

    device-mapper-persistent-data ₩

    lvm2

```
[centos@ip-172-31-23-140 ~]$ sudo yum install -y yum-utils device-mapper-persistent-data lvm2
Loaded plugins: fastestmirror
Determining fastest mirrors
epel/x86_64/metalink                                                    | 7.8 kB  00:00:00
 * base: centos.mirror.cdnetworks.com
 * epel: d2lzkl7pfhq30w.cloudfront.net
 * extras: centos.mirror.cdnetworks.com
 * updates: centos.mirror.cdnetworks.com
base                                                                    | 3.6 kB  00:00:00
epel                                                                    | 3.2 kB  00:00:00
extras                                                                  | 3.4 kB  00:00:00
updates                                                                 | 3.4 kB  00:00:00
epel/x86_64/updateinfo          FAILED
```

4. 설치 및 실행

4.1 설치 진행(CentOS 설치 2/4)

- 레포지토리 추가
  - $ sudo yum-config-manager ₩
    --add-repo ₩ https://download.docker.com/linux/centos/
    docker-ce.repo

```
[centos@ip-172-31-23-140 ~]$ sudo yum-config-manager  --add-repo https://download.docker.com/linux/centos/docker-ce.repo
Loaded plugins: fastestmirror
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
```

4. 설치 및 실행

4.1 설치 진행(CentOS 설치 3/4)

- Docker-ce 설치

  - $ sudo yum install docker-ce

```
[centos@ip-172-31-23-140 ~]$ sudo yum install docker-ce
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: centos.mirror.cdnetworks.com
 * epel: d2lzkl7pfhq30w.cloudfront.net
 * extras: centos.mirror.cdnetworks.com
 * updates: centos.mirror.cdnetworks.com
docker-ce-stable                                                | 2.9 kB  00:00:00
docker-ce-stable/x86_64/primary_db                              |  17 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package docker-ce.x86_64 0:18.06.1.ce-3.el7 will be installed
--> Processing Dependency: container-selinux >= 2.9 for package: docker-ce-18.06.1.ce-3.el7.x86_64
--> Processing Dependency: libltdl.so.7()(64bit) for package: docker-ce-18.06.1.ce-3.el7.x86_64
--> Running transaction check
---> Package container-selinux.noarch 2:2.68-1.el7 will be installed
---> Package libtool-ltdl.x86_64 0:2.4.2-22.el7_3 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================
 Package                    Arch              Version                     Repository                      Size
================================================================================================================
Installing:
 docker-ce                  x86_64            18.06.1.ce-3.el7            docker-ce-stable                41 M
Installing for dependencies:
 container-selinux          noarch            2:2.68-1.el7                extras                          36 k
 libtool-ltdl               x86_64            2.4.2-22.el7_3              base                            49 k

Transaction Summary
================================================================================================================
Install  1 Package (+2 Dependent packages)

Total download size: 41 M
Installed size: 41 M
Is this ok [y/d/N]: y
```

4. 설치 및 실행

4.1 설치 진행(CentOS 설치4/4)

- Docker 버전 확인
  - $ docker version

```
[root@ip-172-31-23-140 ~]# docker version
Client:
 Version:           18.06.1-ce
 API version:       1.38
 Go version:        go1.10.3
 Git commit:        e68fc7a
 Built:             Tue Aug 21 17:23:03 2018
 OS/Arch:           linux/amd64
 Experimental:      false

Server:
 Engine:
  Version:          18.06.1-ce
  API version:      1.38 (minimum version 1.12)
  Go version:       go1.10.3
  Git commit:       e68fc7a
  Built:            Tue Aug 21 17:25:29 2018
  OS/Arch:          linux/amd64
  Experimental:     false
[root@ip-172-31-23-140 ~]#
```

4. 설치 및 실행

4.1 설치 진행(Ubuntu 설치1/6)

- 레포지토리 이용한 설치 방법
  - $ apt-get update 명령어로 apt 패키지 업데이트

```
ubuntu@ip-172-31-29-121:~$ sudo apt-get update
Hit:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic/multiverse Sources [181 kB]
Get:5 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic/restricted Sources [5324 B]
Get:6 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic/main Sources [829 kB]
Get:7 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic/universe Sources [9051 kB]
Get:8 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Get:10 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:11 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:12 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:13 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Sources [200 kB]
Get:14 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Sources [90.8 kB]
Get:15 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Sources [3212 B]
Get:16 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted Sources [2064 B]
Get:17 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [411 kB]
Get:18 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [153 kB]
Get:19 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [7028 B]
Get:20 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [3076 B]
Get:21 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [565 kB]
Get:22 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [148 kB]
Get:23 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [5708 B]
Get:24 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [3176 B]
Get:25 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Sources [1188 B]
Get:26 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [2852 B]
Get:27 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [1200 B]
Get:28 http://security.ubuntu.com/ubuntu bionic-security/main Sources [53.8 kB]
Get:29 http://security.ubuntu.com/ubuntu bionic-security/universe Sources [21.3 kB]
Get:30 http://security.ubuntu.com/ubuntu bionic-security/multiverse Sources [1336 B]
Get:31 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [186 kB]
Get:32 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [72.1 kB]
Get:33 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [89.2 kB]
Get:34 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [48.6 kB]
Get:35 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [1440 B]
Get:36 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [996 B]
Fetched 26.2 MB in 5s (5408 kB/s)
Reading package lists... Done
```

4. 설치 및 실행

4.1 설치 진행(Ubuntu 설치 2/6)

- Apt 통해서 패키지 설치
  - $ sudo apt-get install apt-transport-https ca-certificates curl software-properties-common

```
ubuntu@ip-172-31-29-121:~$ sudo apt-get install apt-transport-https ca-certificates curl software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20180409).
software-properties-common is already the newest version (0.96.24.32.5).
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  curl libcurl4
2 upgraded, 1 newly installed, 0 to remove and 68 not upgraded.
Need to get 374 kB of archives.
After this operation, 152 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 apt-transport-https all 1.6.3ubunt
1696 B]
Get:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 curl amd64 7.58.0-2ubuntu3.3 [159 kB]
Get:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcurl4 amd64 7.58.0-2ubuntu3.3 [214
Fetched 374 kB in 0s (14.9 MB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 56473 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.6.3ubuntu0.1_all.deb ...
Unpacking apt-transport-https (1.6.3ubuntu0.1) ...
Preparing to unpack .../curl_7.58.0-2ubuntu3.3_amd64.deb ...
Unpacking curl (7.58.0-2ubuntu3.3) over (7.58.0-2ubuntu3.2) ...
Preparing to unpack .../libcurl4_7.58.0-2ubuntu3.3_amd64.deb ...
Unpacking libcurl4:amd64 (7.58.0-2ubuntu3.3) over (7.58.0-2ubuntu3.2) ...
Setting up apt-transport-https (1.6.3ubuntu0.1) ...
Setting up libcurl4:amd64 (7.58.0-2ubuntu3.3) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for man-db (2.8.3-2) ...
Setting up curl (7.58.0-2ubuntu3.3) ...
```

4. 설치 및 실행

4.1 설치 진행(Ubuntu 설치 3/6)

- Docker 공식 GPG Key 추가

  - $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –

- Apt-get fingerprint key 확인

  - $ sudo apt-key fingerprint 0EBFCD88

```
ubuntu@ip-172-31-29-121:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
ubuntu@ip-172-31-29-121:~$ sudo apt-key fingerprint 0EBFCD88
pub   rsa4096 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid           [ unknown] Docker Release (CE deb) <docker@docker.com>
sub   rsa4096 2017-02-22 [S]
```

4. 설치 및 실행

4.1 설치 진행(Ubuntu 설치 4/6)

- 레포지토리 추가
  - $ sudo add-apt-repository ₩

    "deb [arch=amd64] https://download.docker.com/linux/ubuntu ₩

    $(lsb_release -cs) ₩

    stable"

```
ubuntu@ip-172-31-29-121:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs)
 stable"
Hit:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Get:4 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Get:5 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [2244 B]
Hit:6 http://security.ubuntu.com/ubuntu bionic-security InRelease
Fetched 66.7 kB in 1s (108 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-29-121:~$ sudo apt-get update
Hit:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:5 http://security.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
```

4. 설치 및 실행

4.1 설치 진행(Ubuntu 설치 5/6)

- Docker-ce 설치

  - $ sudo apt-get install docker-ce

```
ubuntu@ip-172-31-29-121:~$ sudo apt-get install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  aufs-tools cgroupfs-mount libltdl7 pigz
The following NEW packages will be installed:
  aufs-tools cgroupfs-mount docker-ce libltdl7 pigz
0 upgraded, 5 newly installed, 0 to remove and 68 not upgraded.
Need to get 40.4 MB of archives.
After this operation, 199 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 https://download.docker.com/linux/ubuntu bionic/stable amd64 docker-ce amd64 18.06.1~ce~3-0~ubuntu [40.2 MB]
Get:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 aufs-tools amd64 1:4.9+20170918-1ubuntu1 [104 kB
]
Get:4 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 cgroupfs-mount all 1.4 [6320 B]
Get:5 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 libltdl7 amd64 2.4.6-2 [38.8 kB]
Fetched 40.4 MB in 1s (30.3 MB/s)
Selecting previously unselected package pigz.
(Reading database ... 56477 files and directories currently installed.)
Preparing to unpack .../archives/pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package aufs-tools.
Preparing to unpack .../aufs-tools_1%3a4.9+20170918-1ubuntu1_amd64.deb ...
Unpacking aufs-tools (1:4.9+20170918-1ubuntu1) ...
Selecting previously unselected package cgroupfs-mount.
Preparing to unpack .../cgroupfs-mount_1.4_all.deb ...
Unpacking cgroupfs-mount (1.4) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../libltdl7_2.4.6-2_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.6-2) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../docker-ce_18.06.1~ce~3-0~ubuntu_amd64.deb ...
Unpacking docker-ce (18.06.1~ce~3-0~ubuntu) ...
Setting up aufs-tools (1:4.9+20170918-1ubuntu1) ...
Processing triggers for ureadahead (0.100.0-20) ...
Setting up cgroupfs-mount (1.4) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.3) ...
Setting up libltdl7:amd64 (2.4.6-2) ...
Processing triggers for man-db (2.8.3-2) ...
Setting up pigz (2.4-1) ...
Setting up docker-ce (18.06.1~ce~3-0~ubuntu) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for ureadahead (0.100.0-20) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
```

4. 설치 및 실행

4.1 설치 진행(Ubuntu 설치6/6)

- Docker 버전 확인

  - $ docker version

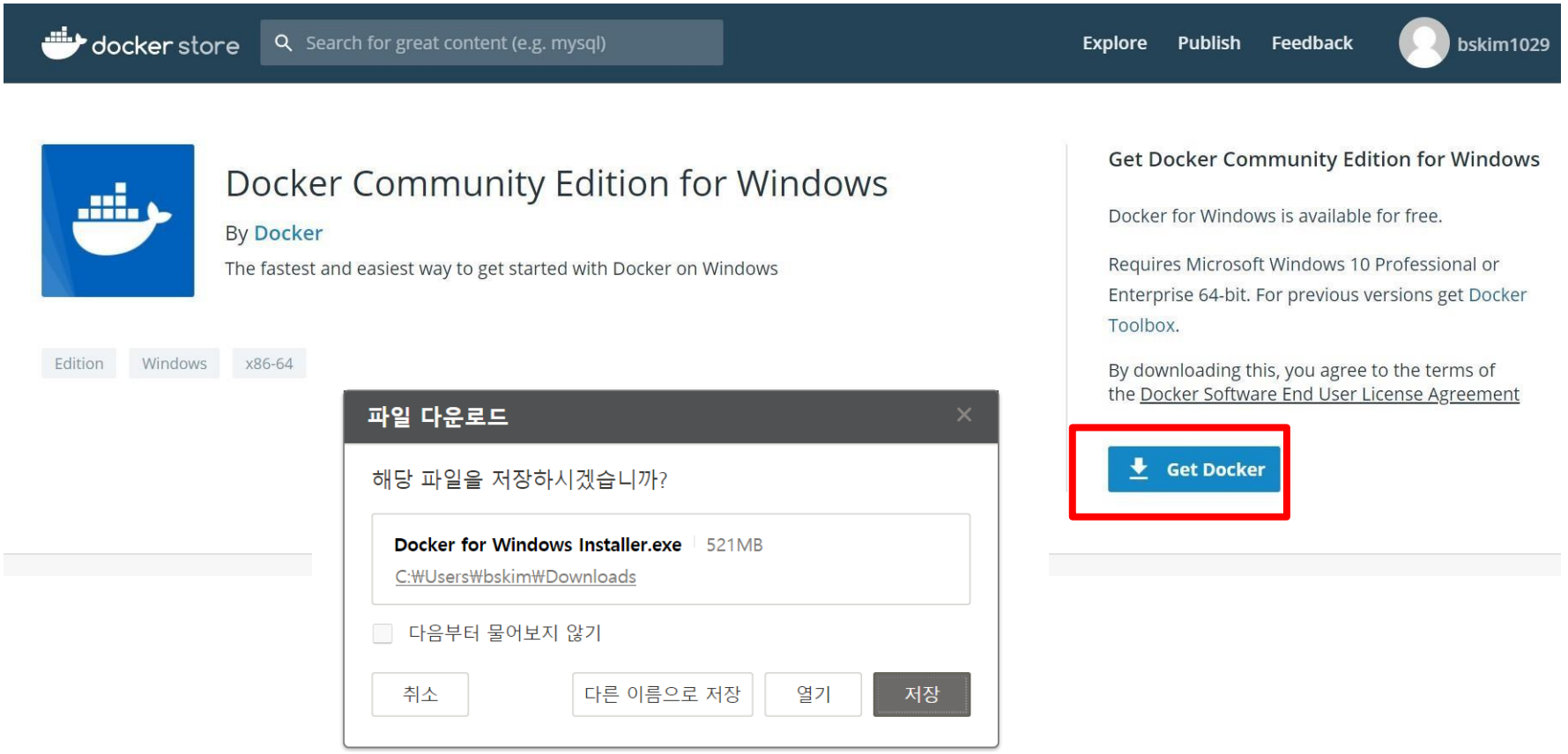
```
root@ip-172-31-29-121:~# docker version
Client:
 Version:           18.06.1-ce
 API version:       1.38
 Go version:        go1.10.3
 Git commit:        e68fc7a
 Built:             Tue Aug 21 17:24:51 2018
 OS/Arch:           linux/amd64
 Experimental:      false

Server:
 Engine:
  Version:          18.06.1-ce
  API version:      1.38 (minimum version 1.12)
  Go version:       go1.10.3
  Git commit:       e68fc7a
  Built:            Tue Aug 21 17:23:15 2018
  OS/Arch:          linux/amd64
  Experimental:     false
root@ip-172-31-29-121:~#
```

4. 설치 및 실행

4.1 설치 진행(Windows 설치1/5)

- Docker store 접속하여 설치 패키지 다운로드

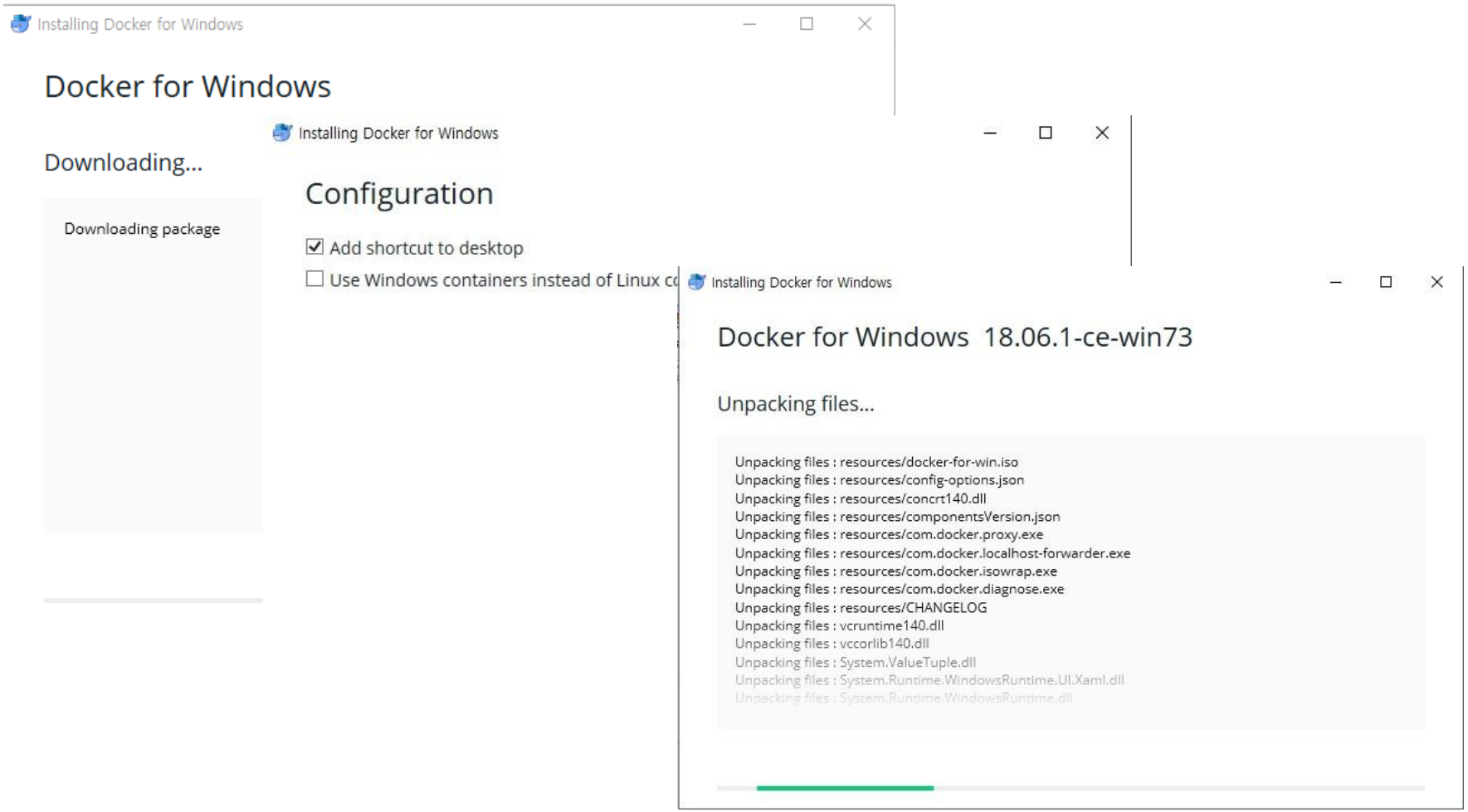
  https://store.docker.com/editions/community/docker-ce-desktop-windows



4. 설치 및 실행

4.1 설치 진행(Windows 설치 2/5)

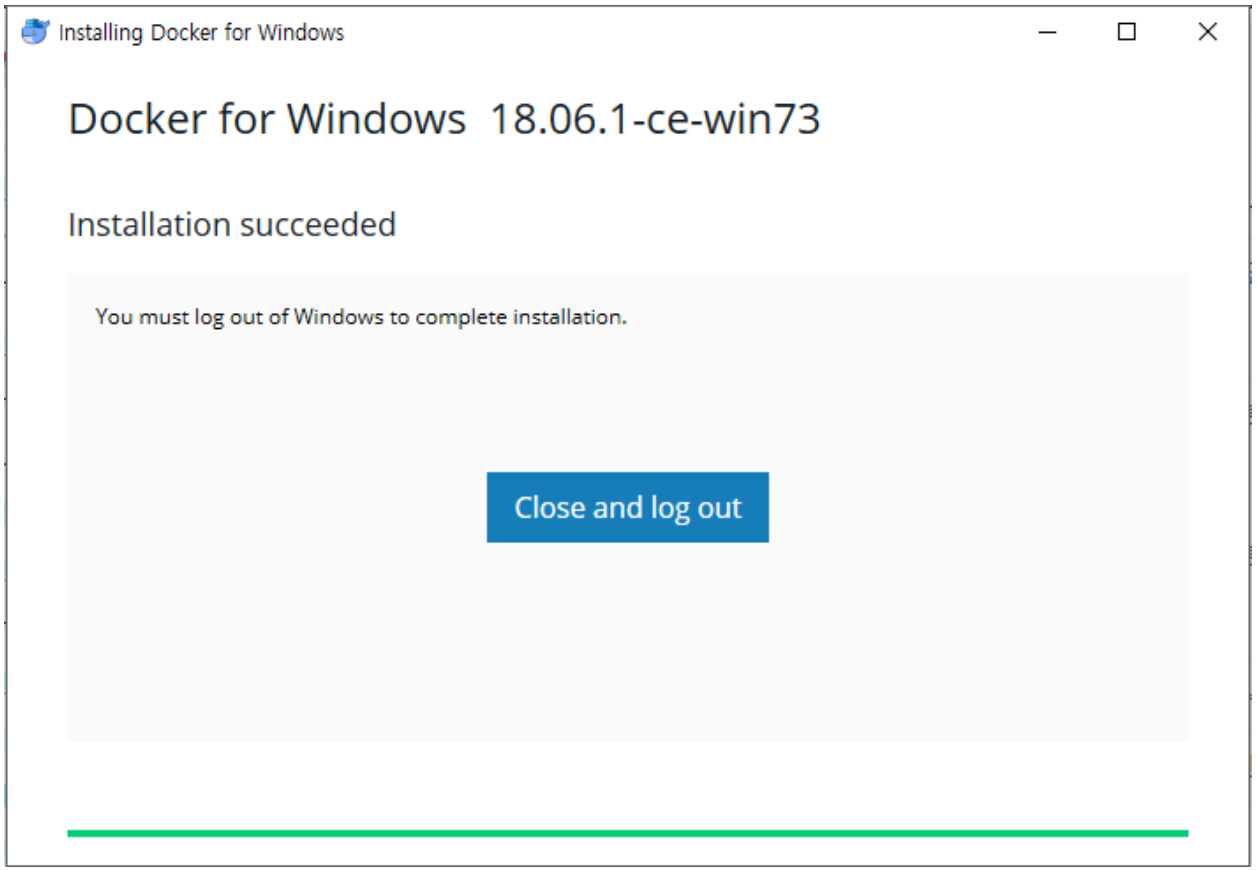
- Docker for Windows Installer.exe 실행하여 설치 진행



4. 설치 및 실행

4.1 설치 진행(Windows 설치 3/5)

- 설치가 끝나면 Close and log out 클릭하여 Windows 로그아웃 진행

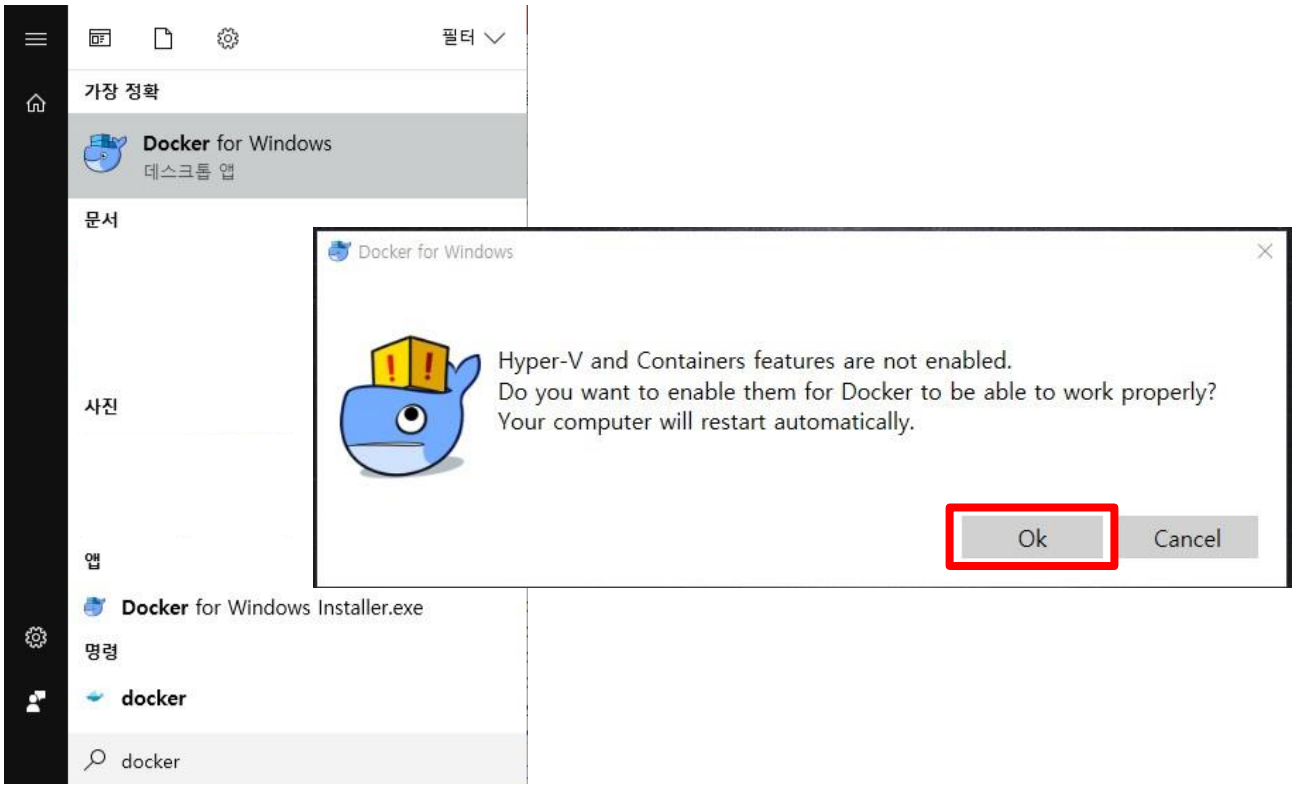


4. 설치 및 실행

4.1 설치 진행(Windows 설치 4/5)

- Windows 검색에서 Docker 찾아 실행

  Windows Hyper-V 사용하고 있지 않았다면 Ok 버튼 클릭

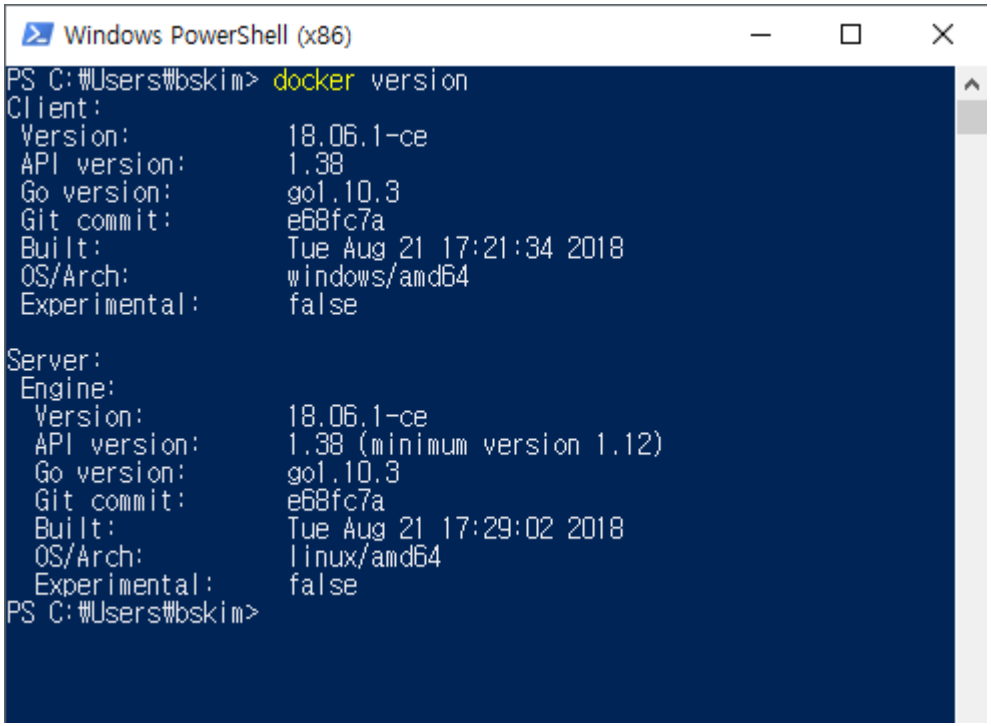


4. 설치 및 실행

4.1 설치 진행(Windows 설치 5/5)

- Windows PowerShell에서 Docker 버전 확인

  - $ docker version



5. 기능소개

세부 목차

1. 컨테이너 실행(run)
2. 컨테이너 목록 확인하기 (ps)
3. 컨테이너 중지하기 (stop)
4. 컨테이너 제거하기 (rm)
5. 이미지 목록 확인하기 (images)
6. 이미지 다운로드하기 (pull)
7. 이미지 삭제하기 (rmi)
8. 컨테이너 로그 보기(logs)
9. 컨테이너에서 명령어 실행(exec)

5. 기능소개

5.1 컨테이너 실행(run 1/4)

- Docker 컨테이너 생성하는 명령어

  docker run <옵션> <이미지 이름, ID> <명령> <매개 변수>

주요 옵션	설명
-a, --attach=[]	표준 입력(stdin), 표준 출력(stdout), 표준 에러(stderr) 연결
--add-host=[]	컨테이너의 /etc/hosts에 호스트 이름과 IP 주소 추가
-c, --cpu-shares=0	CPU 자원 분배 설정, 설정의 기본 값은 1024
--cap-add=[]	cgroups의 특정 Capability 사용
--cap-drop=[]	컨테이너에서 cgroups의 특정 Capability 제외
--cidfile=""	멀티코어 CPU에서 실행될 코어 설정
-d, --detach=false	Detached 모드
--device=[]	호스트의 장치 사용할 수 있도록 연결

5. 기능소개

5.1 컨테이너 실행(run 2/4)

주요 옵션	설명
--dns=[]	DNS 서버 설정
--dns-search=[]	DNS 검색 도메인 설정
-e, --env=[]	환경 변수 설정
--entrypoint=""	Dockerfile의 ENTRYPOINT 무시하고 다른 값 설정
--env-file=[]	환경 변수가 설정된 파일 적용
--expose=[]	포트 호스트와 연결
-h, --hostname=""	호스트 이름 설정
-i, --interactive=false	표준 입력(stdin) 활성화, 컨테이너 표준 입력 유지
--link=[]	컨테이너간 연결
--lxc-conf=[]	LXC 옵션

5. 기능소개

5.1 컨테이너 실행(run 3/4)

주요 옵션	설명
-m, --memory=""	메모리 한계 설정
--name=""	컨테이너 이름 설정
--net=""	네트워크 모드 설정
-P, --publish-all=false	호스트에 연결된 컨테이너 포트 외부노출
-p, --publish=[]	호스트에 연결된 컨테이너 특정 포트 외부노출
--privileged=false	컨테이너에서 호스트의 리눅스 커널 기능 사용
--restart=""	컨테이너 프로세스 종료 시 재시작 정책 설정
--rm=false	프로세스 종료되면 컨테이너 자동삭제
--security-opt=[]	SELinux, AppArmor 옵션 설정
--sig-proxy=true	모든 시그널 프로세스에 전달

5. 기능소개

5.1 컨테이너 실행(run 4/4)

주요 옵션	설명
-t, --tty=false	TTY 모드(pseudo-TTY) 사용
-u, --user=""	리눅스 사용자 계정 이름/UID 설정
-v, --volume=[]	데이터 볼륨 설정
--volumes-from=[]	데이터 볼륨 컨테이너 연결
-w, --workdir=""	프로세스가 실행될 디렉터리 설정

5. 기능소개

5.2 컨테이너 목록 확인하기 (ps)

- 컨테이너 목록 확인하는 명령어

  docker ps <옵션>

주요 옵션	설명
a, --all=false	모든 컨테이너 출력
--before=""	특정 컨테이너가 생성되기 전에 생성된 컨테이너 출력, 정지된 컨테이너도 포함
-f, --filter=[]	출력 필터 설정 예) "exited=0"
-l, --latest=false	마지막에 생성된 컨테이너 출력 정지된 컨테이너도 포함
-n=-1	최근에 생성된 컨테이너 일정 개수만 출력 정지된 컨테이너도 포함
--no-trunc=false	생략된 부분 모두 출력
-q, --quiet=false	컨테이너 ID 출력

5. 기능소개

5.3 컨테이너 중지하기 (stop)

- 실행중인 컨테이너 중지하는 명령어

  docker stop <옵션> CONTAINER [CONTAINER...]

주요 옵션	설명
-t, –time=10	컨테이너 정지하기 전 대기하는 시간 설정 (초 단위)

5. 기능소개

5.4 컨테이너 제거하기 (rm)

- 컨테이너 제거하는 명령어

  docker rm <옵션> <컨테이너 이름, ID>

주요 옵션	설명
-f, --force=false	컨테이너 강제로 정지한 뒤 삭제(SIGKILL 시그널 사용)
-l, --link=false	docker run 명령의 --link 옵션 사용하여 연결된 링크만 삭제
-v, --volumes=false	컨테이너에 연결된 데이터 볼륨 삭제

5. 기능소개

5.5 이미지 목록 확인하기 (images)

- 도커 이미지 목록 출력하는 명령어

  docker images <옵션> <이미지 이름>

주요 옵션	설명
-a, --all=false	상속 이미지까지 모두 표시
-f, --filter=[]	출력 결과 필터 설정 (“dangling=true” 이름이 없는 이미지만 출력)
--no-trunc=false	생략된 부분 모두 출력
-q, --quiet=false	이미지 ID만 출력

5. 기능소개

5.6 이미지 다운로드하기 (pull)

- Docker 레지스트리에서 이미지 받아오는 명령어

  docker pull <옵션> <저장소 이름>/<이미지 이름>:<태그>

주요 옵션	설명
-a, --all=false	이미지의 모든 태그 받음

5. 기능소개

5.7 이미지 삭제하기 (rmi)

- 이미지 삭제하는 명령어, 만약 태그 지정하지 않으면 latest 태그 삭제

  docker rmi <저장소 이름>/<이미지 이름, ID>:<태그>

주요 옵션	설명
-f, --force=false	이미지 강제로 삭제
--no-prune=false	태그가 없는 상속 이미지 삭제하지 않음

5. 기능소개

5.8 컨테이너 로그 보기(logs)

- 컨테이너의 로그 출력하는 명령어

  docker logs <컨테이너 이름, ID>

주요 옵션	설명
-f, --follow=false	로그 실시간 출력
-t, --timestamps=false	로그 앞에 시간 값 표시
--tail="all"	숫자 지정하여 최종 로그에서 일정 개수만 출력

5. 기능소개

5.9 컨테이너에서 명령어 실행(exec)

- 외부에서 컨테이너 안의 명령 실행하는 명령어

  docker exec <옵션> <컨테이너 이름, ID> <명령> <매개 변수>

주요 옵션	설명
-d, --detach=false	명령 백그라운드로 실행
-i, --interactive=false	표준 입력(stdin) 활성화하며 컨테이너와 연결(attach)되어 있지 않더라도 표준 입력 유지
-t, --tty=false	TTY 모드(pseudo-TTY) 사용 (Bash 사용시 필요, 명령 입력은 가능하지만 쉘이 표시되지 않음)

6. 활용예제

세부 목차

6.1 Nginx docker container 서비스

6. 활용예제

6.1 Nginx docker container 서비스(1/2)

- Docker 이용하여 Nginx 웹서버 구축

  dockerrun-d-p80:80–read-only-v$(pwd)/nginx-cache:/var/cache/nginx-v$(pwd)/nginx-pid:/var/runnginx

```
root@ip-172-31-29-121:~# docker run -d -p 80:80 --read-only -v $(pwd)/nginx-cache:/var/cache/nginx -v $(pwd)/nginx-pid:/var/run nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
f17d81b4b692: Pull complete
d5c237920c39: Pull complete
a381f92f36de: Pull complete
Digest: sha256:b73f527d86e3461fd652f62cf47e7b375196063bbbd503e853af5be16597cb2e
Status: Downloaded newer image for nginx:latest
f44cd1461f9fc1e4c1f769edf6a58e4a63dc5cb82bbc53be41a2566788aaf64d
root@ip-172-31-29-121:~#
```

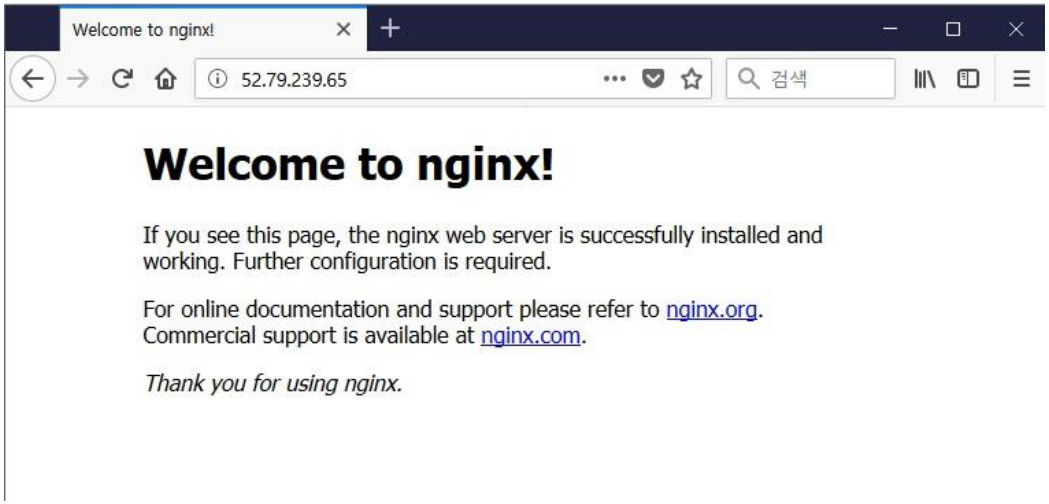
- Nginx 컨테이너가 호스트 포트 80으로 서비스되고 있는 것 볼 수 있음

```
root@ip-172-31-29-121:~# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
f44cd1461f9f        nginx               "nginx -g 'daemon of…"   11 seconds ago      Up 9 seconds        0.0.0.0:80->80/tcp  frosty_liskov
fb271c358680        ubuntu:16.04        "/bin/bash -c 'while…"   4 hours ago         Up 4 hours                              hello
root@ip-172-31-29-121:~#
root@ip-172-31-29-121:~#
root@ip-172-31-29-121:~# netstat -nap | grep 80
tcp6       0      0 :::80                   :::*                    LISTEN      31531/docker-proxy
unix  2      [ ACC ]     STREAM     LISTENING     62931    14143/docker-contai  @/containerd-shim/moby/fb271c358680cf60c6f59a2bfac3c1485fe3c95013d
025250e1e8ff9d763e240/shim.sock@
unix  3      [ ]         DGRAM                    12380    1/init               /run/systemd/notify
unix  3      [ ]         STREAM     CONNECTED     74680    31372/sshd: ubuntu@
unix  3      [ ]         STREAM     CONNECTED     62937    14143/docker-contai  @/containerd-shim/moby/fb271c358680cf60c6f59a2bfac3c1485fe3c95013d
025250e1e8ff9d763e240/shim.sock@
root@ip-172-31-29-121:~#
```

6. 활용예제

6.1 Nginx docker container 서비스(2/2)

- 웹브라우저에서 Nginx 서비스 호출



7. FAQ

Q **Docker란 무엇입니까?**

A Docker는 애플리케이션 신속하게 구축, 테스트 및 배포할 수 있는 소프트웨어 플랫폼입니다. Docker는 소프트웨어 컨테이너라는 표준화된 유닛으로 패키징하며, 이 컨테이너에는 라이브러리, 시스템 도구, 코드, 런타임 등 소프트웨어 실행하는데 필요한 모든 것이 포함되어 있습니다.

Q **Docker 사용해야 하는 이유가 있을까요?**

A Docker 사용하면 코드 더 빨리 전달하고, 애플리케이션 운영 표준화하고, 코드 원활하게 이동하고, 리소스 사용률 높여 비용 절감할 수 있습니다.

8. 용어정리

용어	설 명
Docker Hub	도커 관련 중앙저장소, 도커 이미지 호스팅, 사용자 인증, 자동 빌드 기능
docker image	컨테이너의 기초가 되는 이미지
tag	저장소 내에서 도커 이미지에 부여하는 꼬리표
Nginx	웹 서버 소프트웨어로, 가벼움과 높은 성능
Dockerfile	빌드 방법, 명령어 써놓은 텍스트 문서
container	도커 이미지의 런타임 인스턴스

Open Source Software Installation & Application Guide