docker

도커 무작정 따라하기

도커가 처음인 사람도 60분이면 웹 서버를 올릴 수 있습니다!

PYRASIS.COM 이재홍

version 1.0.3, 2015.2.9

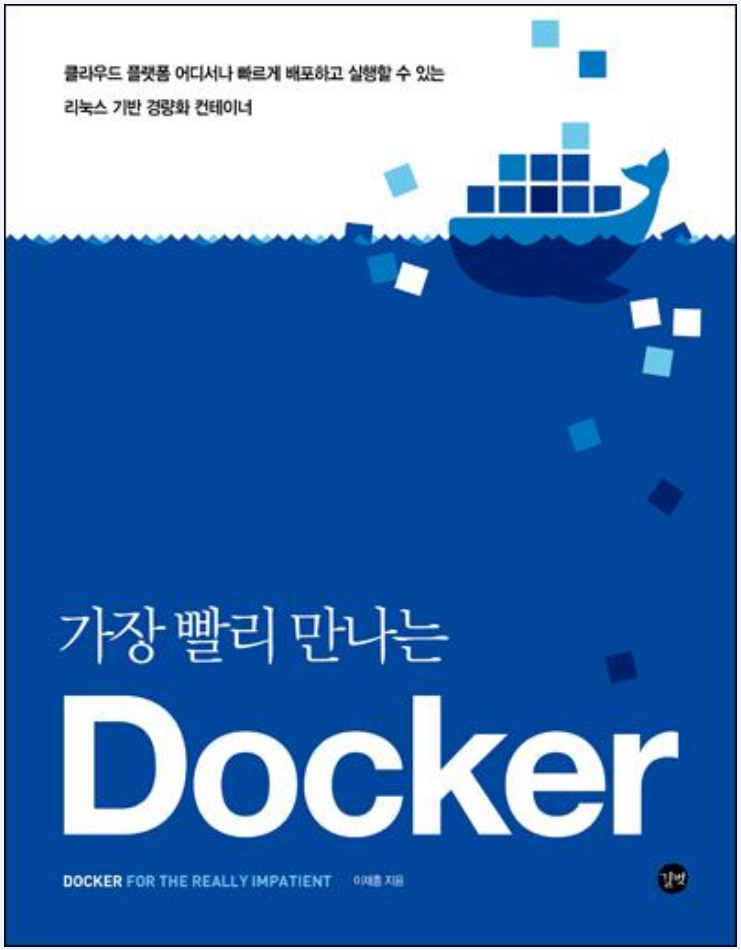
문서 이용 조건

교육 등의 비영리 목적으로만 사용해야 합니다. 사내 교육은 상관 없습니다.

회비, 참가비 등을 받는다면 영리 목적에 해당하므로 이 문서를 이용할 수 없습니다.

회비, 참가비를 받지 않는다 하더라도 마케팅 목적으로 진행되는 행사에는 이 문서를 사용할 수 없습니다.

만든 사람 및 슬라이드 소개



이재홍

웹사이트: http://pyrasis.com

저서:

- ✓ 가장 빨리 만나는 도커(Docker)
- ✓ 아마존 웹 서비스를 다루는 기술
- ✓ 윈도우 프로젝트 필수 유틸리티: Subversion, Trac, CruiseControl.NET

이 슬라이드는 가장 빨리 만나는 도커의 내용을 요약한 슬라이드입니다.

가장 빨리 만나는 도커의 책 내용은 모두 웹사이트에 공개되어 있습니다.

도커

도커란 무엇인가?

도커는 2013년 3월 Docker, Inc에서 출시한

오픈 소스 컨테이너 프로젝트

현재 전세계적으로 큰 인기를 끌고 있음

도커란 무엇인가?

도커

AWS, Google cloud Platform, Microsoft Azure 등의

클라우드 서비스에서 공식 지원

도커란 무엇인가?

도커

도커는 왜 인기를 끌게 되었나?

도커란 무엇인가?

복잡한 리눅스 어플리케이션을
컨테이너로 묶어서 실행할 수 있음

도커란 무엇인가?

도커

개발, 테스트, 서비스 환경을 하나로 통일하여
효율적으로 관리할 수 있기 때문!

도커란 무엇인가?

컨테이너(이미지)를

전 세계 사람들과 공유

도커란 무엇인가?

도커

그건 어떻게 가능하죠?

도커란 무엇인가?

리눅스 커널에서 제공하는

컨테이너 기술을 이용

도커란 무엇인가?

GitHub와 비슷한 방식의

Docker Hub 제공!

도커란 무엇인가?

도커

그런데 컨테이너 기술이 뭐죠?

도커란 무엇인가?

도커

컨테이너는 가상화보다

훨씬 가벼운 기술!

도커란 무엇인가?

도커

그럼 먼저 먼저 가상화부터...

가상 머신과 도커

가상 머신의 등장

가상 머신과 도커

컴퓨터 안에서 컴퓨터를 만들어내기 위한 시도

1960년 대에 가상화 개념이 처음 등장

가상 머신의 등장

컴퓨터 성능이 급격히 좋아지면서

PC에서도 흔히 사용

맥에서 인터넷 뱅킹을

한다던지...

가상 머신의 등장

가상 머신과 도커

서버 성능은 더욱 더 좋아졌음

가상 머신의 등장

가상 머신과 도커

그러다 보니 대부분의 시간을

서버가 놀고 있어 -_-;

가상 머신의 등장

가상 머신과 도커

서버에 가상 머신을 여러 개

띄워서 일을 더 시키자!

그리고 IT 기술이 보편화되면서

서버도 많아졌음

가상 머신의 등장

가상 머신과 도커

서버 자체를 가상 머신에
집어넣어서 돌리자

가상 머신의 등장

가상 머신과 도커

가상 머신에 각종 서버 프로그램, DB 등을 설치하여
애플리케이션이나 웹사이트를 실행

가상 머신의 등장

가상 머신과 도커

미리 구축한 가상 머신 이미지를

여러 서버에 복사하여 실행하면

이미지 하나로 서버를 계속 만들어낼 수 있음

가상 머신의 등장

가상 머신과 도커

가상화 기술을 이용하여 서버를
임대해주는 서비스가 클라우드 서비스

가상 머신의 문제점

가상 머신과 도커

그런데 가상 머신은 좀 문제가 있어

가상 머신의 문제점

가상 머신과 도커

컴퓨터를 통째로 만들어내다 보니...

각종 성능 손실이 발생

가상 머신의 문제점

가상 머신과 도커

인텔과 AMD는 CPU 안에 가상화 기능을 넣기 시작

가상 머신의 문제점

가상 머신과 도커

그래도 느려...

호스트와 커널을 공유하는 반가상화 기술이 등장

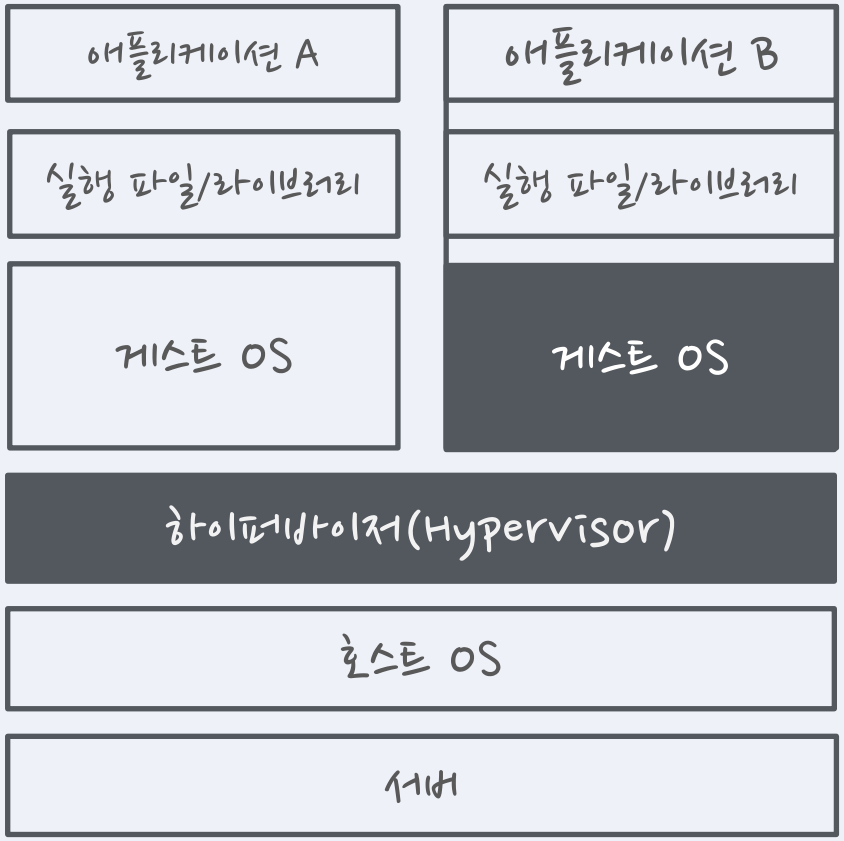
가상 머신의 문제점

가상 머신과 도커

이러나 저러나 가상 머신은...

가상 머신의 문제점

가상 머신과 도커



https://www.docker.com/whatisdocker/

가상 머신은 완전한 컴퓨터

✓항상 게스트 OS를 설치해야 함

가상 머신의 문제점

가상 머신과 도커

이미지 안에 OS가 포함되기 때문에 이미지 용량이 커짐

✓ 네트워크로 가상화 이미지를 주고 받는 건 꽤 부담스러움

가상 머신의 문제점

가상 머신과 도커

오픈소스 가상화 소프트웨어는 OS 가상화에 만 주력

✓ 배포와 관리 기능이 부족

가상 머신의 문제점

가상 머신과 도커

가상 머신의 성능 문제가 있다 보니

리눅스 컨테이너가 나옴

리눅스 컨테이너

가상 머신과 도커

컨테이너 안에 가상 공간을 만들지만

실행 파일을 호스트에서 직접 실행

이건 리눅스 커널의

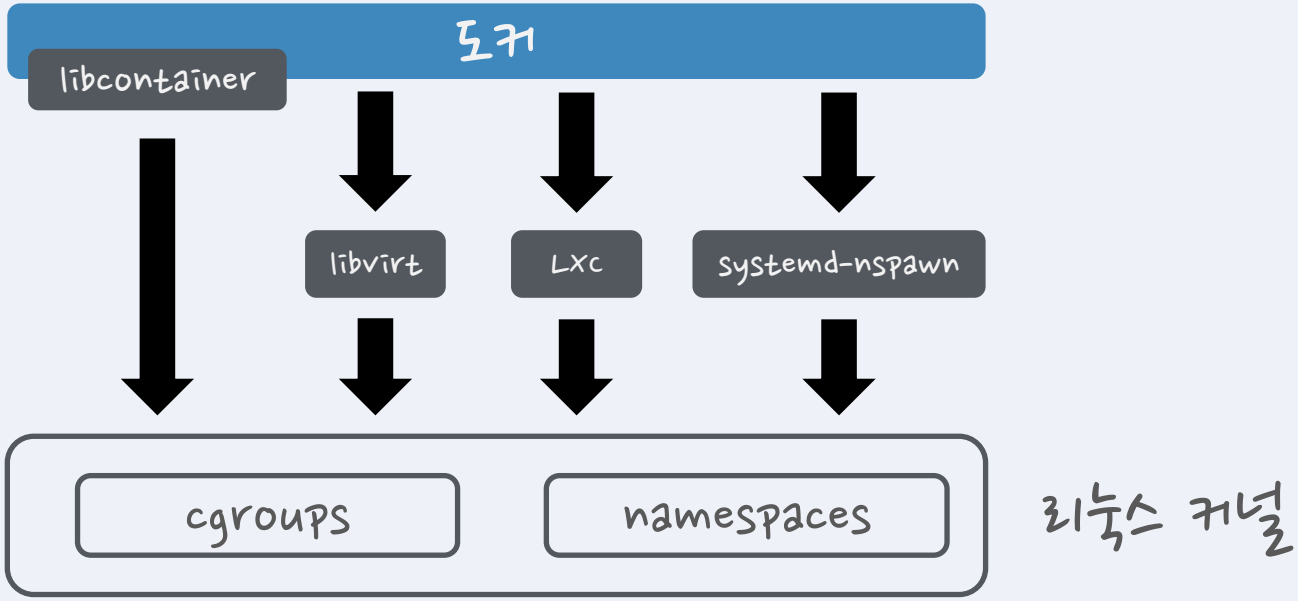
cgroups와 namespaces가

제공하는 기술

가상화가 아닌 격리

리눅스 컨테이너

가상 머신과 도커



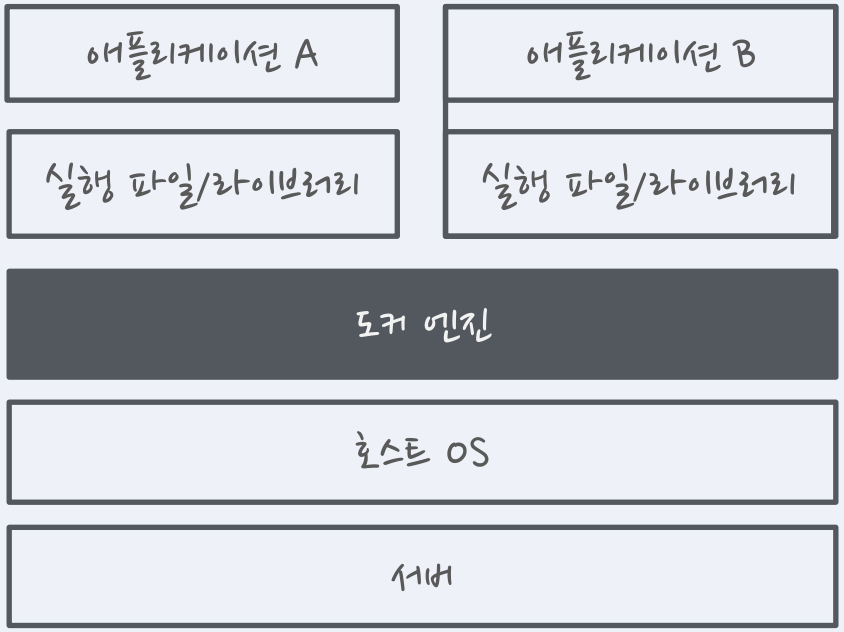
http://blog.docker.com/2014/03/docker-0-9-introducing-execution-drivers-and-libcontainer

도커는 리눅스 컨테이너를 사용!

- ✓ 초기에는 LXC(Linux container)를 기반으로 구현
- ✓ 버전 0.9부터는 LXC를 대신하는 libcontainer를 개발하여 사용
- ✓ 실행 옵션으로 선택 가능

도커의 특징

도커의 특징



https://www.docker.com/whatisdocker/

도커는 게스트 OS를 설치하지 않음

✓ 이미지에 서버 운영을 위한 프로그램과 라이브러리만 격리해서 설치

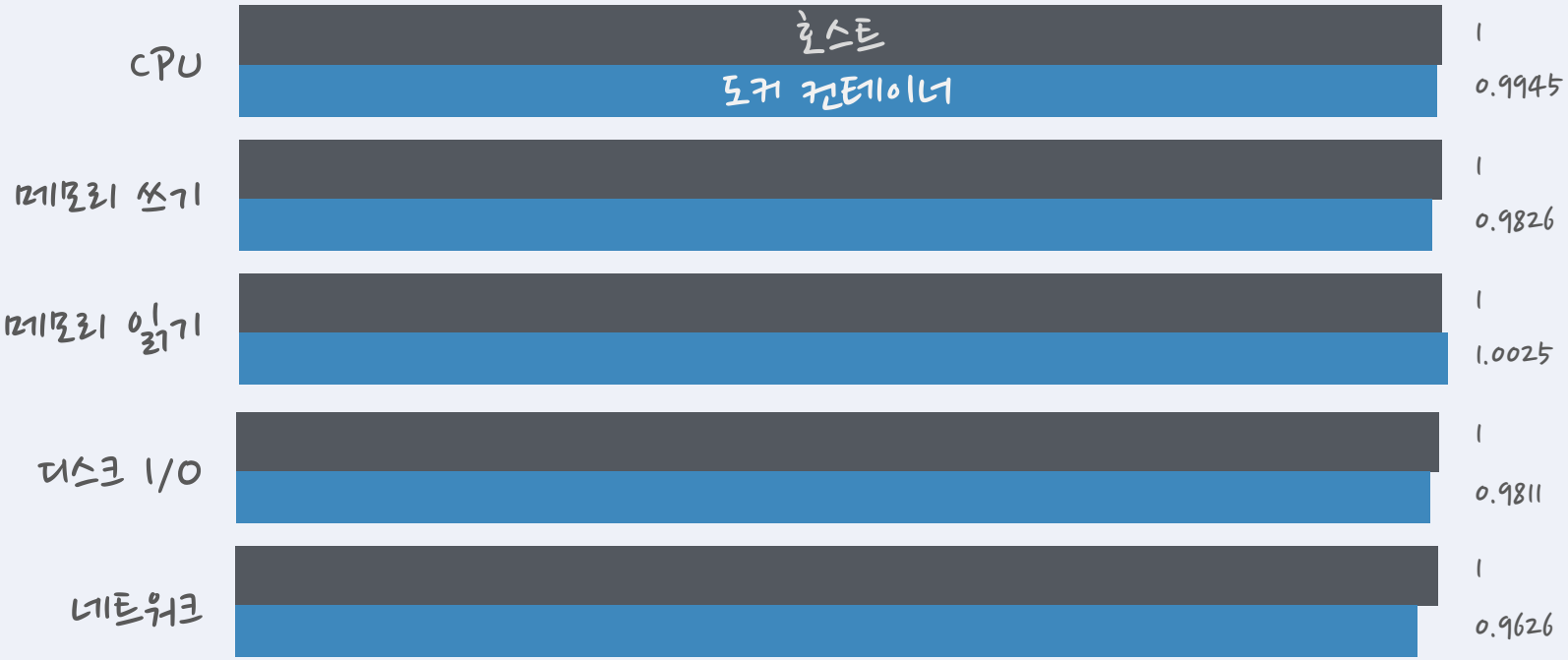
✓ 이미지 용량이 크게 줄어듦

✓ 호스트와 OS 자원(시스템 콜)을 공유

도커의 성능

도커의 특징

도커 1.1.2에서 우분투 14.04 호스트와 우분투 14.04 컨테이너 성능 측정



도커는 하드웨어 가상화 계층이 없음

✓ 메모리 접근, 파일 시스템, 네트워크 전송 속도가 가상 머신에 비해 월등히 빠름

✓ 호스트와 도커 컨테이너 사이의 성능 차이가 크지 않음(오차 범위 안)

그리고 도커는

이미지 생성과 배포에 특화

도커의 특징

이미지 버전 관리도 제공하고

중앙 저장소에 이미지를 올리고 받을 수 있음

(push/pull)

도커의 특징

GitHub와 비슷한 형태로

도커 이미지를 공유하는 Docker Hub 제공

(GitHub 처럼 유료 저장소도 제공)

도커의 특징

다양한 API를 제공하여 원하는 만큼 자동화 가능

개발과 서버 운영에 매우 유용!

그런데 이미지와 컨테이너는

뭐가 다른 거지?

도커 이미지와 컨테이너

도커 이미지

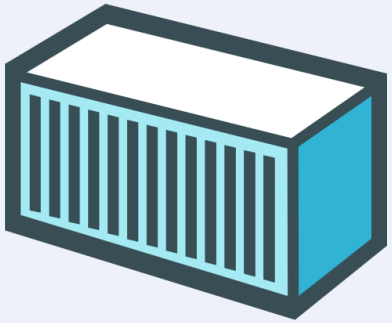
도커 이미지와 컨테이너

이미지는 서비스 운영에 필요한 서버 프로그램,
소스 코드, 컴파일된 실행 파일을 묶은 형태

저장소에 올리고 받는건 이미지 (push/pull)

도커 컨테이너

도커 이미지와 컨테이너



컨테이너는 이미지를 실행한 상태!

도커 컨테이너

이미지로 여러 개의 컨테이너를 만들 수 있음

운영체제로 치면 이미지는 실행파일이고

컨테이너는 프로세스

도커의 이미지 처리 방식

도커는 이미지의 바뀐 부분을

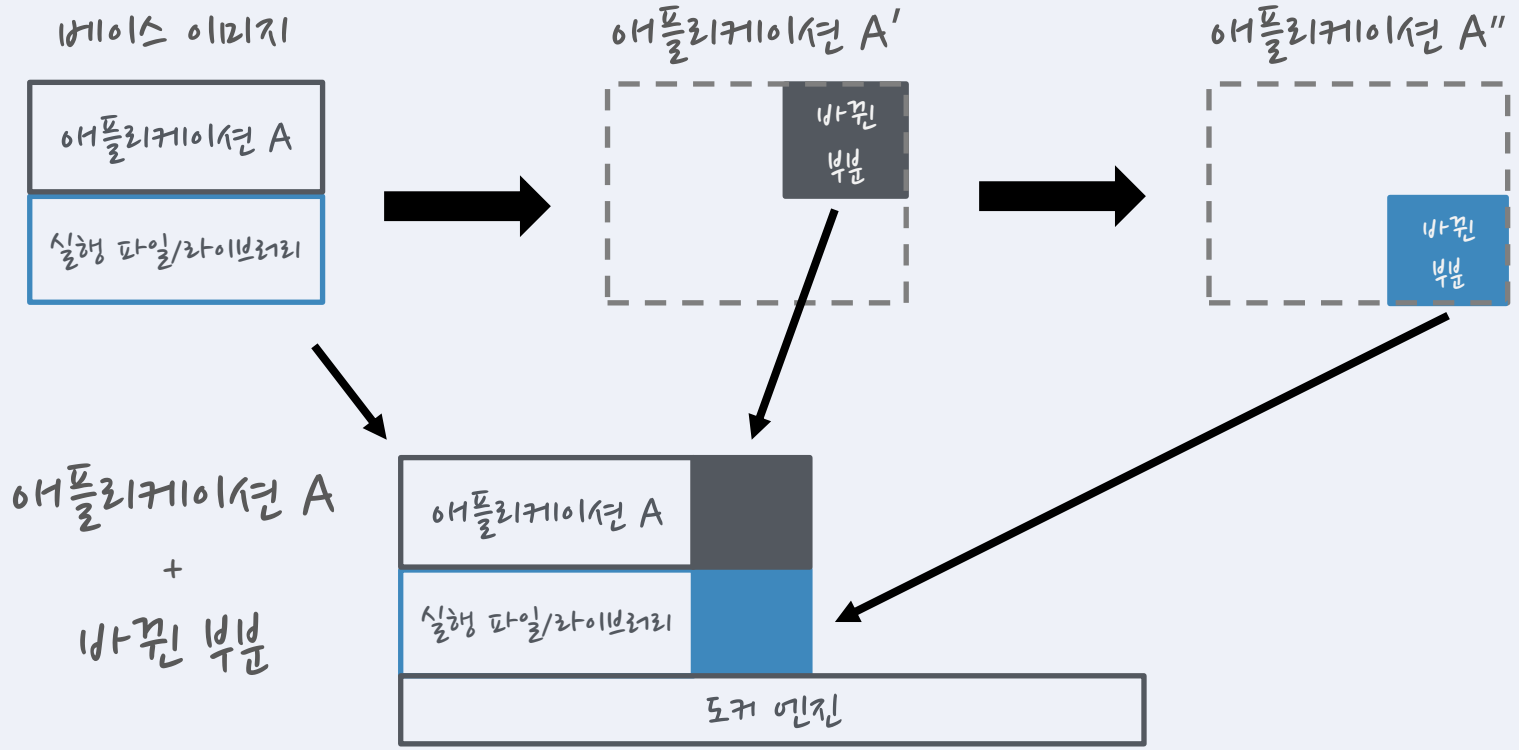
어떻게 관리하나?

도커의 이미지 처리 방식

유니온 파일 시스템 형식
(aufs, btrfs, devicemapper)

도커의 이미지 처리 방식

도커 이미지와 컨테이너



http://www.slideshare.net/dotCloud/why-docker

도커는 베이스 이미지에서 바뀐 부분만 이미지로 생성

컨테이너를 실행할 때는 베이스 이미지와 바뀐 부분을 합쳐서 실행

도커의 이미지 처리 방식

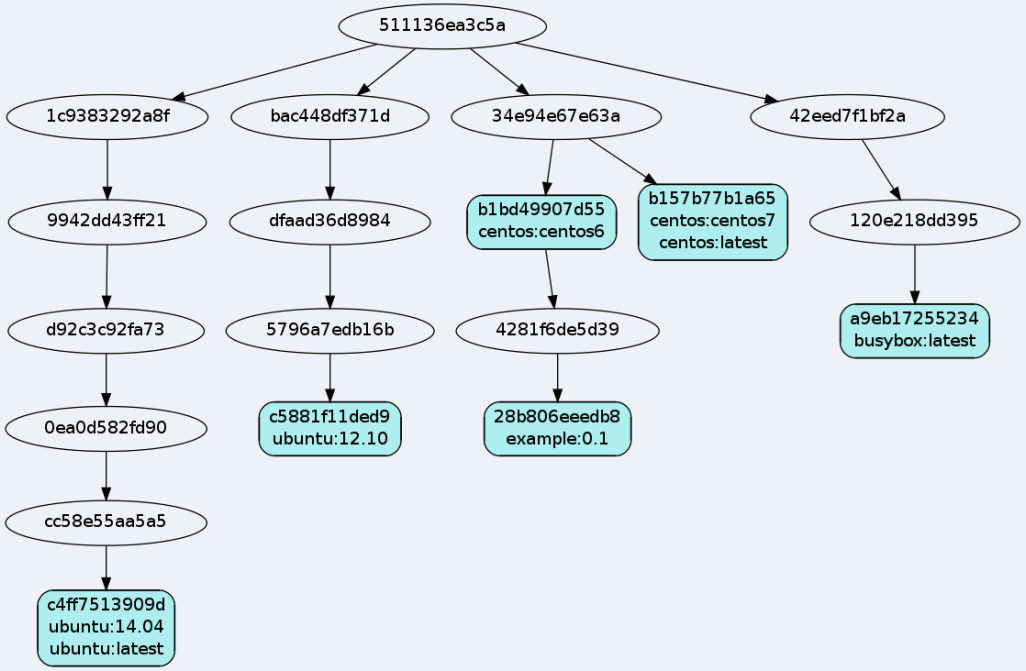
Docker Hub 및 개인 저장소에서

이미지를 공유할 때

바뀐 부분만 주고 받음

도커의 이미지 처리 방식

도커 이미지와 컨테이너



각 이미지는 의존 관계 형성

서비스 운영 환경과 도커

지금까지의 서버 환경

서비스 운영 환경과 도커

지금까지는

물리 서버를 직접 운영 했음

호스팅 또는 IDC 코로케이션 서비스 사용

지금까지의 서버 환경

서비스 운영 환경과 도커

서버 구입과 설치에 돈이 많이 들고

시간이 오래 걸림

클라우드 환경

서비스 운영 환경과 도커

가상화가 발전하면서

클라우드 환경으로 변화

클라우드 환경

서비스 운영 환경과 도커

가상 서버를 임대하여

사용한 만큼만 요금 지불

클라우드 환경

서비스 운영 환경과 도커

클릭 몇 번 만으로 가상 서버를 생성

클라우드 환경

서비스 운영 환경과 도커

이전엔 자동으로 서버를 추가하고

삭제하기까지...

클라우드 환경

서비스 운영 환경과 도커

서버 대수가 많아지면서

사람이 일일이 세팅하기 힘들어짐

이제 서버 세팅과 배포는 어떻게?

Immutable Infrastructure

서비스 운영 환경과 도커

Immutable Infrastructure라는

패러다임이 나옴

Immutable Infrastructure

서비스 운영 환경과 도커

호스트 OS와 서비스 운영 환경

(서버 프로그램, 소스 코드, 컴파일 된 바이너리)을 분리

Immutable Infrastructure

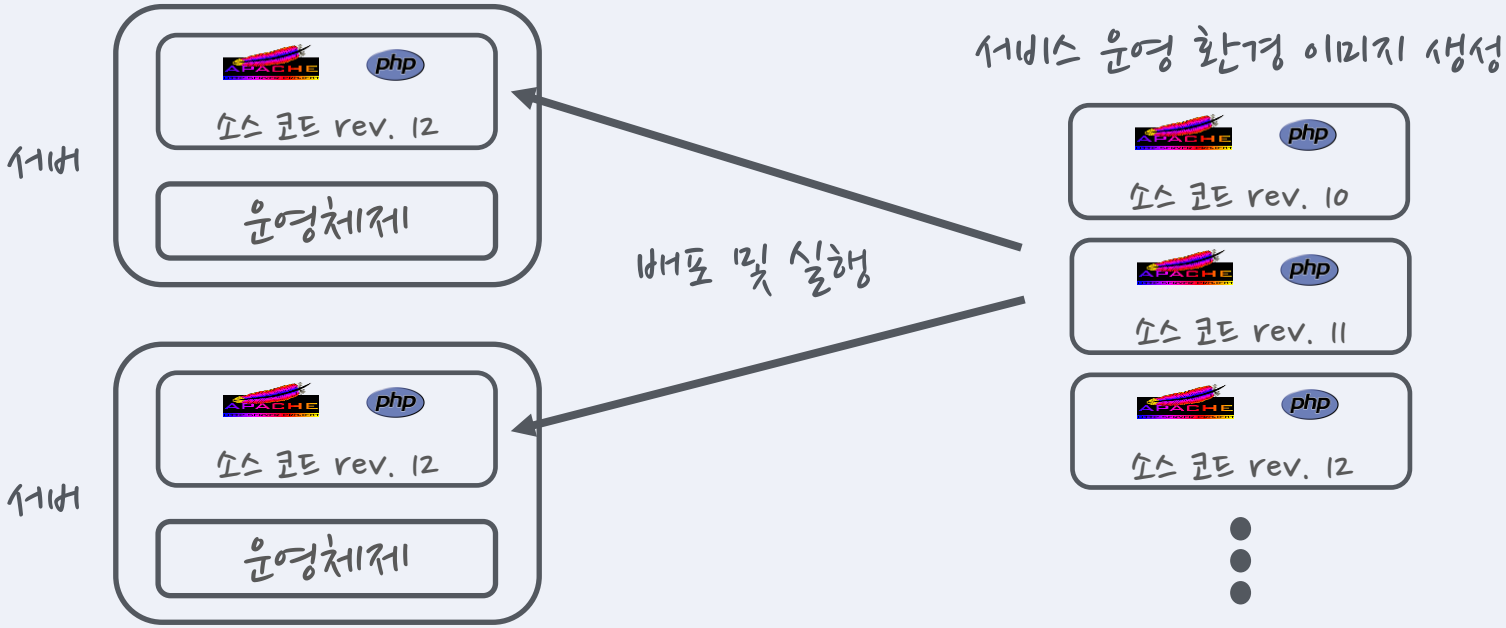
서비스 운영 환경과 도커

한 번 설정한 운영 환경은

변경하지 않는다(Immutable)는 개념

Immutable Infrastructure

서비스 운영 환경과 도커



서비스 운영 환경을 이미지로 생성한 뒤

서버에 배포하여 실행

Immutable Infrastructure

서비스 운영 환경과 도커

서비스가 업데이트되면 운영 환경 자체를 변경하지 않고, 이미지를 새로 생성하여 배포

Immutable Infrastructure

서비스 운영 환경과 도커

클라우드 플랫폼에서 서버를 쓰고 버리는 것과 같이

Immutable Infrastructure도

서비스 운영 환경 이미지를 한 번 쓰고 버림

Immutable Infrastructure의 장점

서비스 운영 환경과 도커

편리한 관리

- ✓ 서비스 환경 이미지만 관리하면 됨
- ✓ 중앙 관리를 통한 체계적인 배포와 관리
- ✓ 이미지 생성에 버전 관리 시스템 활용

확장

- ✓ 이미지 하나로 서버를 계속 찍어낼 수 있음
- ✓ 클라우드 플랫폼의 자동 확장(Auto Scaling) 기능과 연동하여 손쉽게 서비스 확장

테스트

- ✓ 개발자 PC, 테스트 서버에서 이미지를 실행만 하면 서비스 운영 환경과 동일한 환경이 구성됨
- ✓ 테스트가 간편

가볍다

- ✓ 운영체제와 서비스 환경을 분리하여 가볍고(Lightweight) 어디서든 실행 가능한(Portable) 환경 제공

Immutable Infrastructure

서비스 운영 환경과 도커

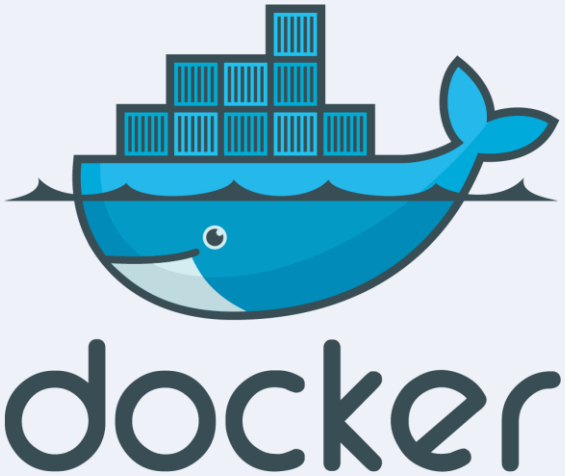
도커는 Immutable Infrastructure를
구현한 프로젝트

도커 요약

도커 고래 로고 많이들 보셨죠?

로고로 알아보는 도커

도커 요약



컨테이너를 싣고 다니는 고래

도커로 알아보는 도커

도커 요약

고래는 서버에서 여러 개의 컨테이너(이미지)를 실행하고

이미지 저장과 배포(운반)을 의미

로고로 알아보는 도커

도커 요약

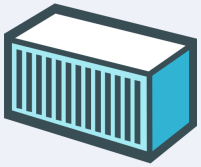
도커(Docker)는 부두 노동자를 뜻함

컨테이너를 다루는 도커의 기능과 비슷함

도커를 요약하면

도커 요약

Build



Ship

Run

도커는 서비스 운영 환경을 묶어서

손쉽게 배포하고 실행하는

경량 컨테이너 기술

도커 설치하기

도커 설치하기

이제 자신의 컴퓨터에서 명령을 실행하여
직접 도커를 설치하고 사용해봅니다.

자동 설치 스크립트

도커 설치하기: 리눅스

✓ 도커는 리눅스 배포판 종류를 자동으로 인식하여 도커 패키지를 설치해주는 스크립트를 제공

```
$ sudo wget -qO- https://get.docker.com/ | sh
```

get.docket.com 스크립트로 도커를 설치하면 hello-world 이미지도 자동 설치됨.

Hello-world 이미지는 사용하지 않을 것이므로 모두 삭제

```
$ sudo docker rm `sudo docker ps -aq`
$ sudo docker rmi hello-world
```

우분투

도커 설치하기: 리눅스

✓ 자동 설치 스크립트를 사용하지 않고 우분투에서 직접 설치하기.

버전은 14.04 LTS 64비트 기준

```
$ sudo apt-get update
$ sudo apt-get install docker.io
$ sudo ln -sf /usr/bin/docker.io /usr/local/bin/docker
```

/usr/bin/docker.io 실행 파일을 /usr/local/bin/docker로 링크하여 사용

RedHat Enterprise Linux, centOS

도커 설치하기: 리눅스

- ✓ 자동 설치 스크립트를 사용하지 않고 레드햇 엔터프라이즈 리눅스(RHEL)와 centOS에서 패키지로 직접 설치하기
- ✓ RHEL과 centOS 6 패키지 저장소에는 docker-io가 없으므로 EPEL(Fedora Extra Packages For Enterprise Linux) 저장소를 사용할 수 있도록 설정

CentOS 6

```
$ sudo yum install http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
$ sudo yum install docker-io
```

RedHat Enterprise Linux, centOS

도커 설치하기: 리눅스

- ✓ AWS EC2에 설치되는 Amazon Linux(RHEL 기반)는 EPEL 저장소를 바로 사용할 수 있으므로 epel-release-6-8.noarch.rpm을 설치하지 않아도 됨
- ✓ centOS 7에서는 docker 패키지를 설치

CentOS 7

```
$ sudo yum install docker
```

Docker 서비스 실행하기

```
$ sudo service docker start
```

부팅했을 때 자동으로 실행하기

```
$ sudo chkconfig docker on
```

최신 바이너리 사용하기

도커 설치하기: 리눅스

✓ 배포판 버전이 오래되었거나, CentOS 같이 버전업이 보수적인 배포판에서는 도커 패키지 버전이 낮은 경우가 있음

✓ 배포판 별 패키지가 아닌 빌드 된 바이너리를 직접 사용하는 방법

이미 패키지로 설치했을 때

```
$ sudo service docker stop
$ sudo wget https://get.docker.com/builds/Linux/x86_64/docker-latest \
  -O $(type -P docker)
$ sudo service docker start
```

최신 바이너리 사용하기

도커 설치하기: 리눅스

새로 설치할 때

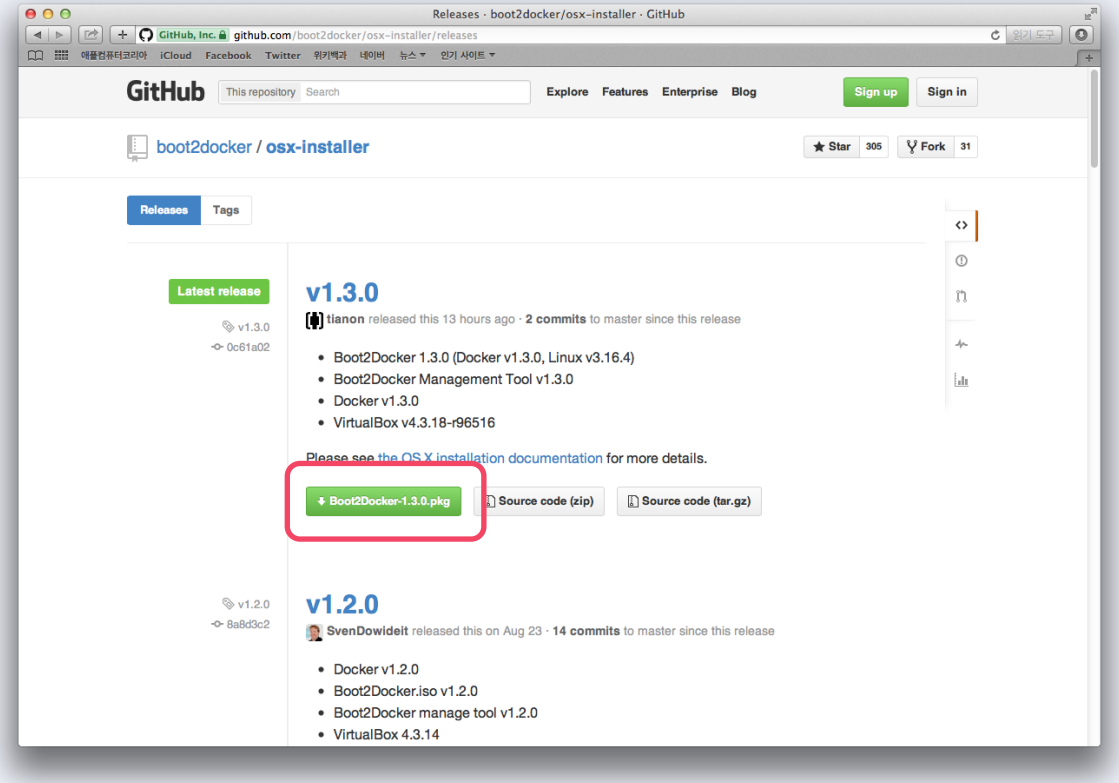
```
$ wget https://get.docker.com/builds/Linux/x86_64/docker-latest
$ chmod +x docker-latest
$ sudo mv docker-latest /usr/local/bin/docker
$ sudo /usr/local/bin/docker -d
```

URL을 docker-latest로 지정하면 가장 최신 버전을 받고, docker-1.3.0처럼 지정하면 특정 버전을 받음

Mac OS X

도커 설치하기

- ✓ Mac OS X에서는 Boot2Docker를 이용하여 도커를 사용할 수 있음
- ✓ 다음 URL에서 .pkg 파일을 받기
  - ✓ https://github.com/boot2docker/osx-installer/releases



Mac OS X

도커 설치하기

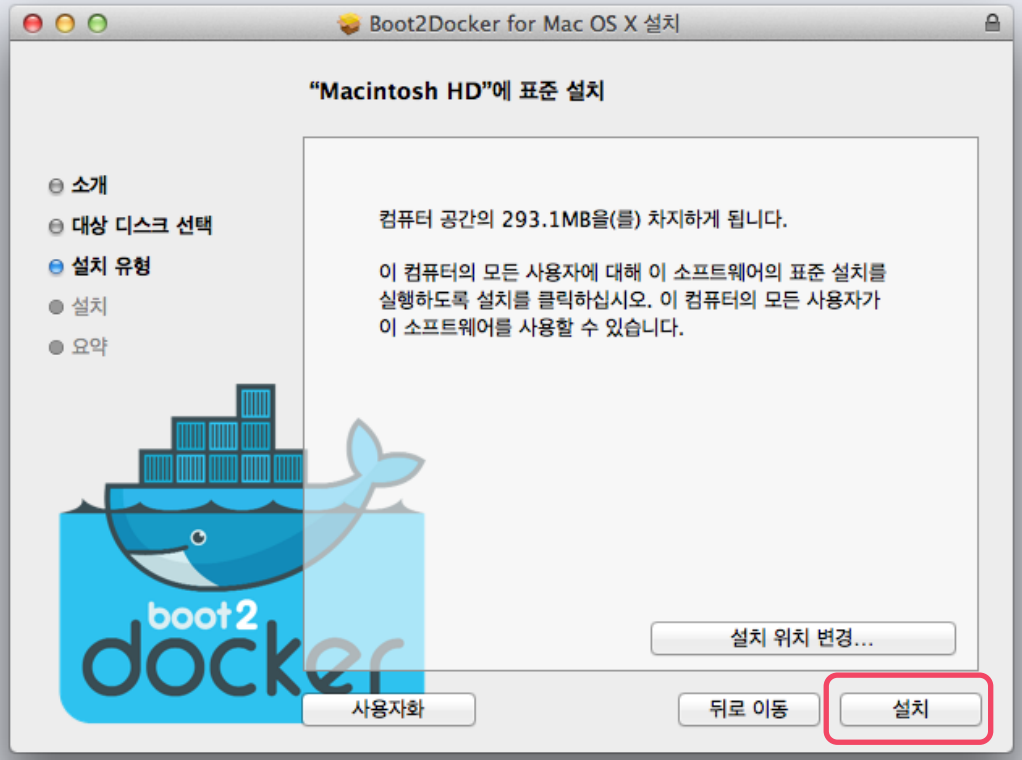
✓ 파일 다운로드가 끝났으면 .pkg 파일 실행

✓ 설치화면이 표시되면 계속 버튼을 클릭

Mac OS X

도커 설치하기

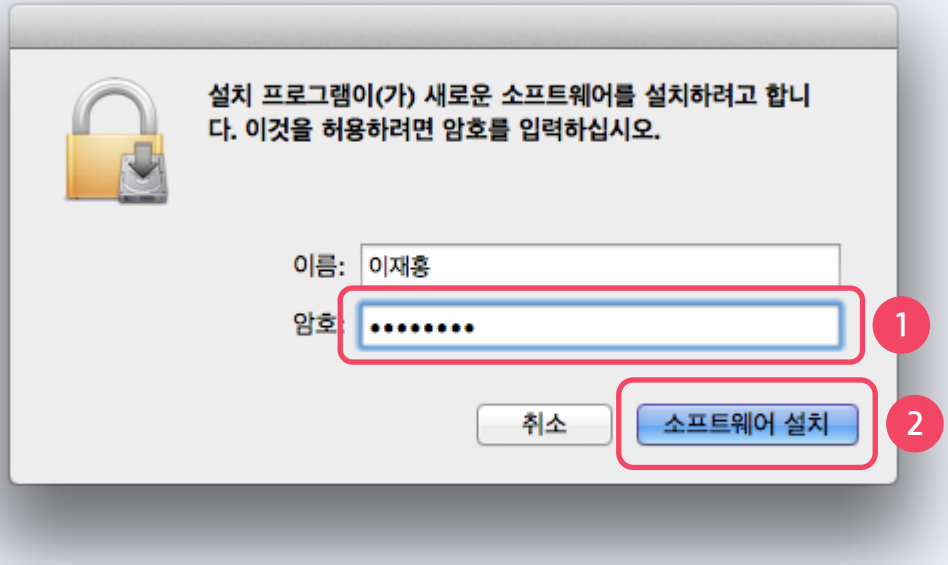
✓ 기본 위치에 설치



Mac OS X

도커 설치하기

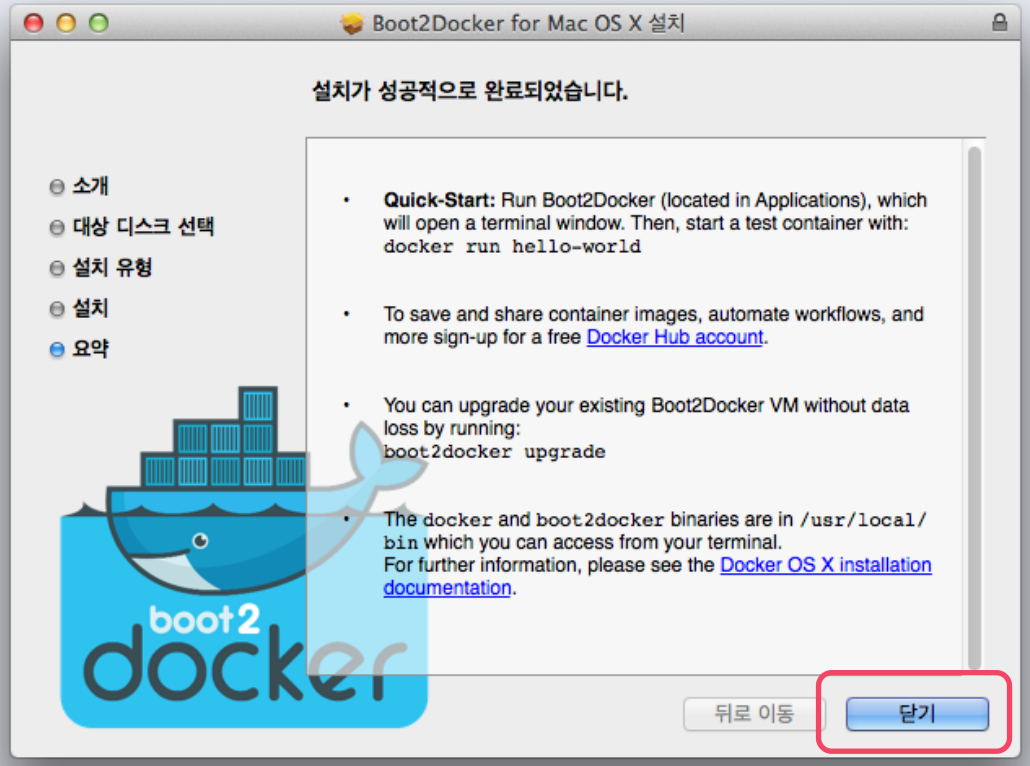
✓ 암호 입력창이 표시되면 관리자 암호를 입력하고 소프트웨어 설치 버튼을 클릭



Mac OS X

도커 설치하기

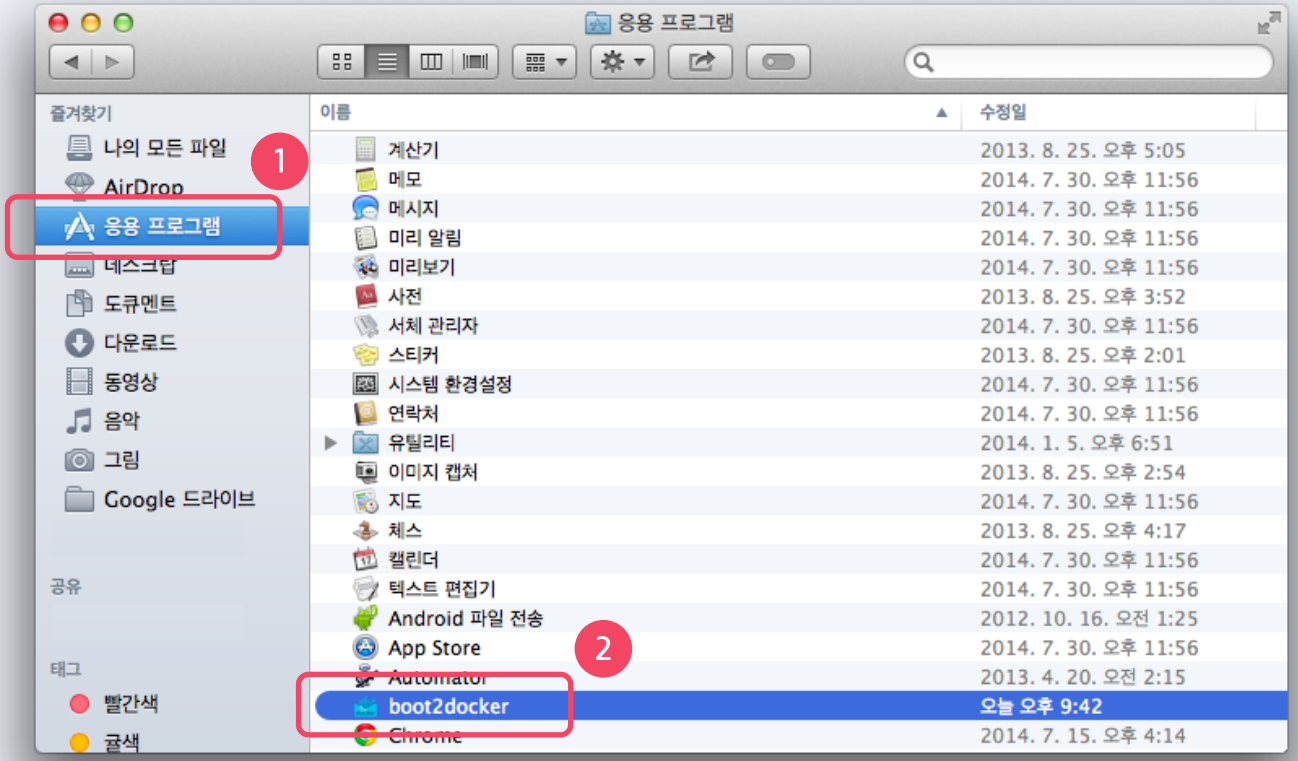
✓ 설치가 완료되었으면 닫기 버튼 클릭



Mac OS X

도커 설치하기

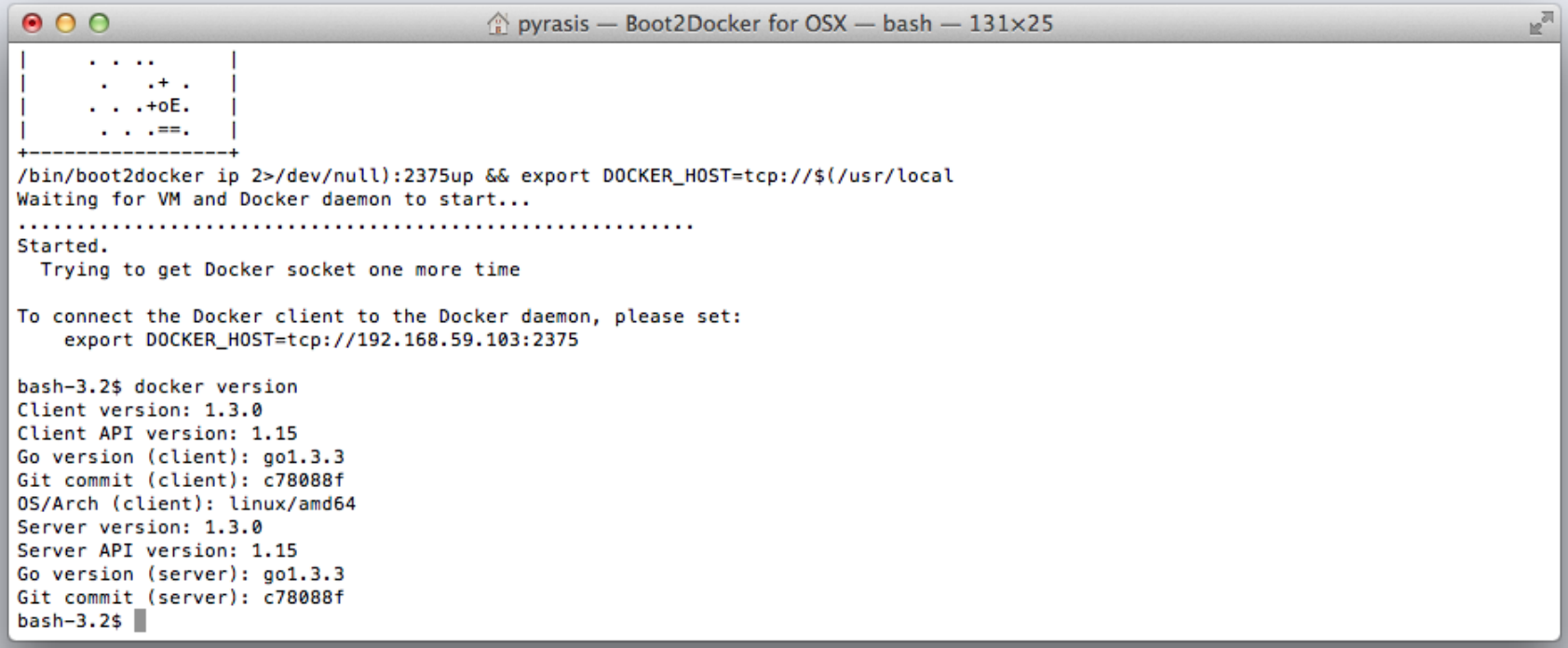
✓ Finder를 실행하고 응용 프로그램에서 boot2docker를 실행



Mac OS X

도커 설치하기

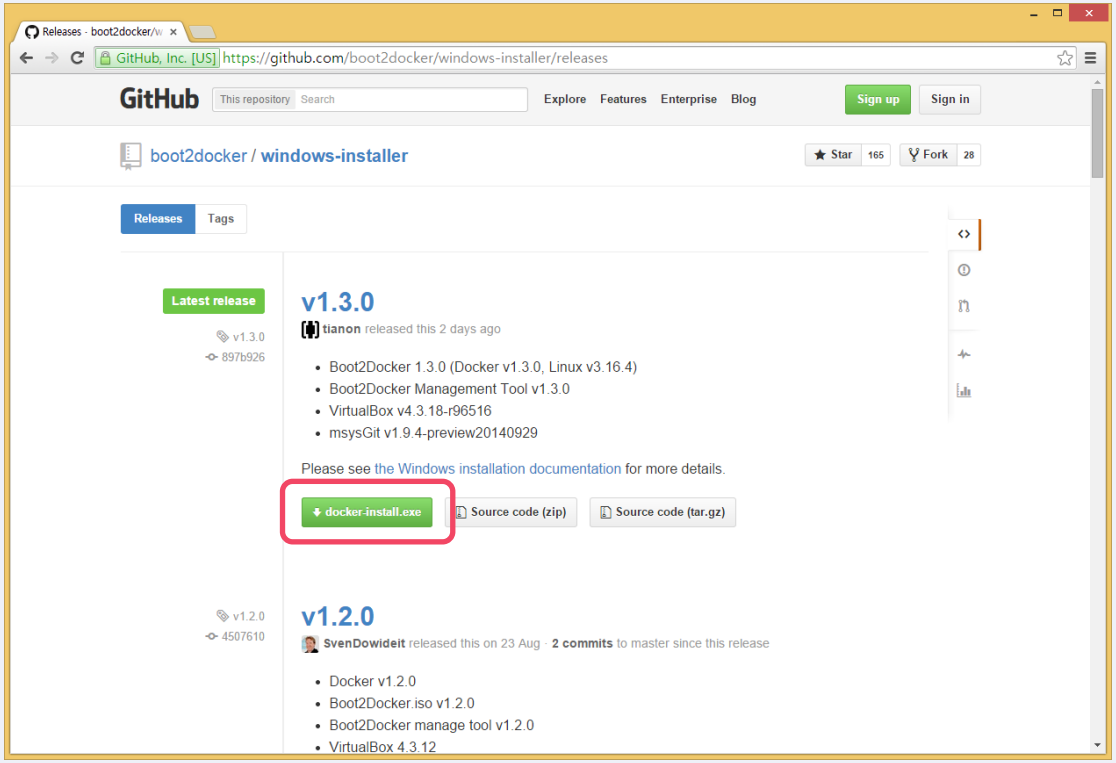
- ✓ boot2docker를 실행하면 터미널 창이 실행됨
- ✓ 잠시 기다리면 자동으로 boot2docker.iso를 이용하여 가상 머신이 생성되고, 가상 머신에 접속됨(Boot2Docker는 내부적으로 VirtualBox가 함께 설치됨)



Windows

도커 설치하기

- ✓ Windows에서는 Boot2Docker를 이용하여 Docker를 사용할 수 있음
- ✓ 다음 URL에서 docker-install.exe 파일을 받기
  - ✓ https://github.com/boot2docker/windows-installer/releases

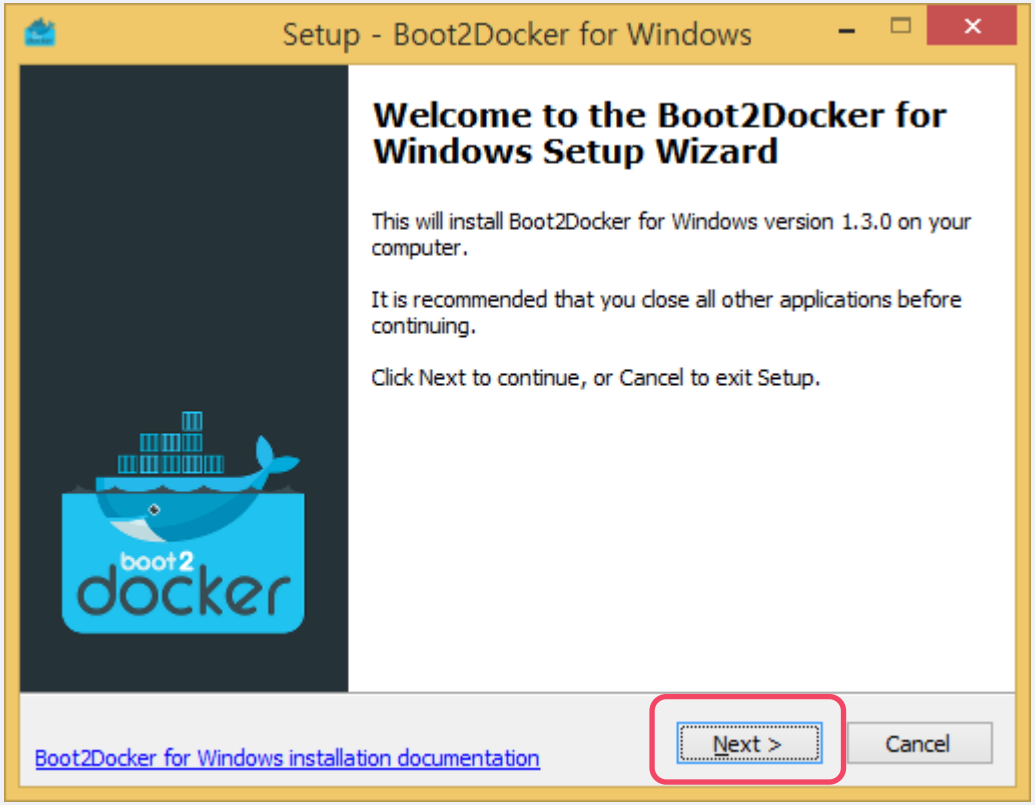


Windows

도커 설치하기

✓ 파일 다운로드가 끝났으면 docker-install.exe 파일을 실행

✓ 설치 화면이 표시되면 Next 버튼을 클릭

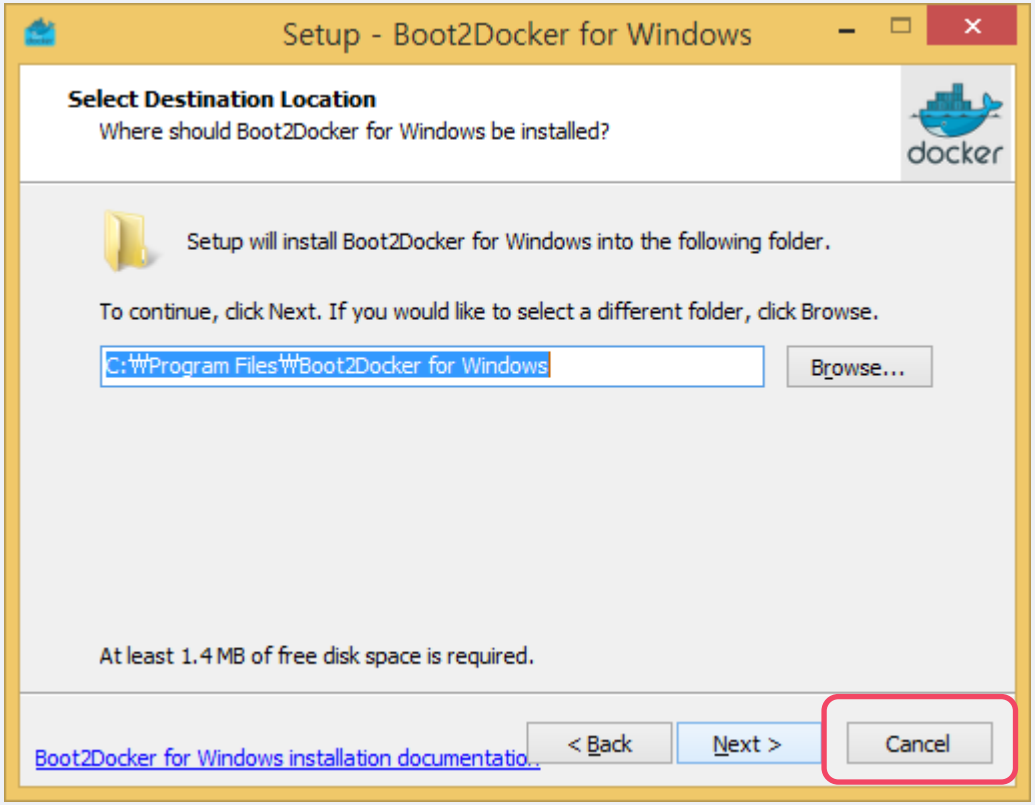


Windows

도커 설치하기

✓ 기본 위치에 설치

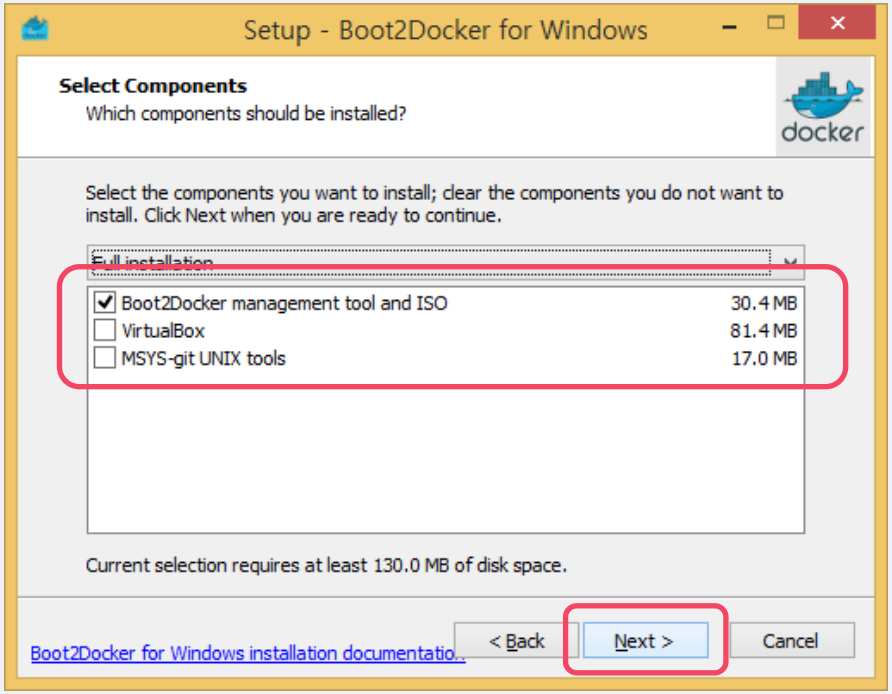
✓ Next 버튼 클릭



Windows

도커 설치하기

- ✓ 설치할 요소 선택
  - ✓ Boot2Docker Management script and ISO: Boot2Docker를 설치할 것이므로 반드시 선택
  - ✓ VirtualBox: VirtualBox를 설치하지 않았다면 이 부분을 선택
  - ✓ MSYS-git UNIX tools: Windows용 Git을 설치하지 않았다면 이 부분을 선택
- ✓ 선택이 완료되었으면 Next 버튼을 클릭

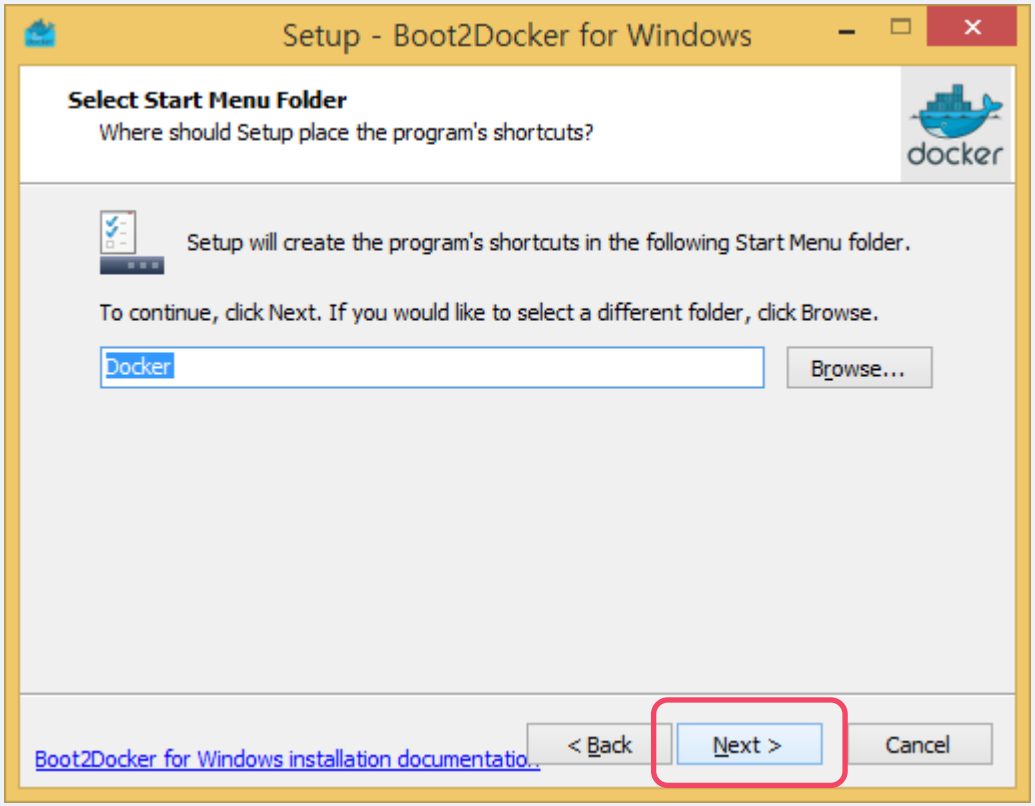


Windows

도커 설치하기

✓ 시작 메뉴에 등록할 이름 설정

✓ 기본 값 그대로 사용하고 Next 버튼을 클릭

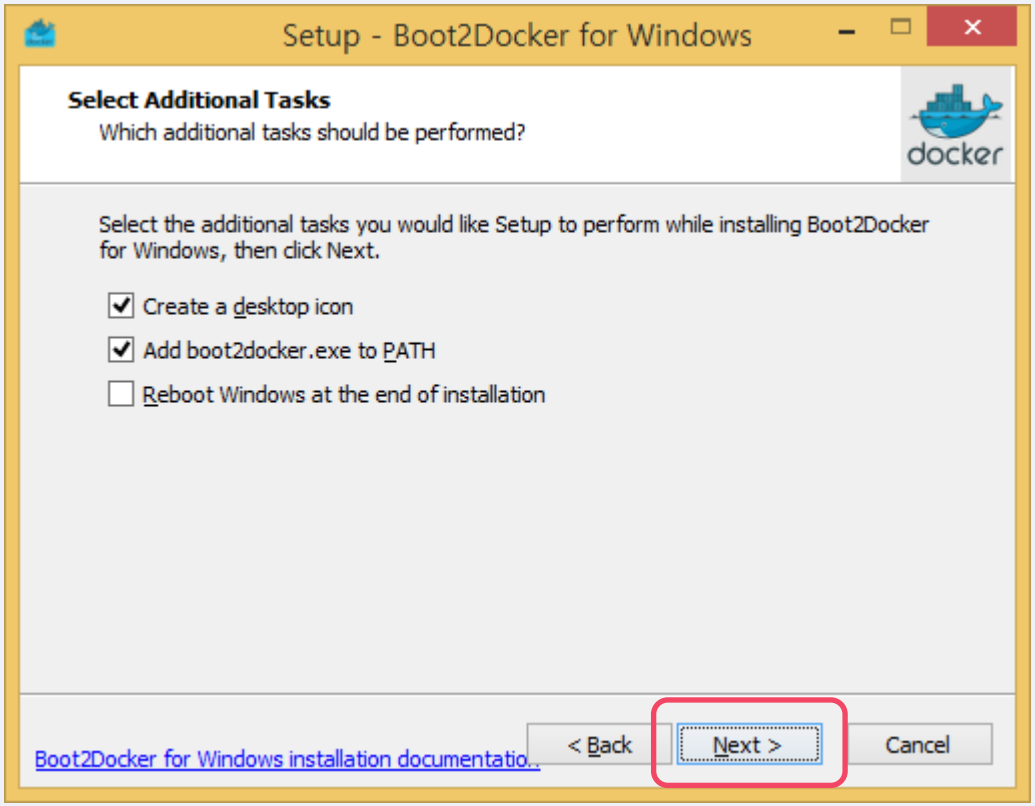


Windows

도커 설치하기

✓ Boot2Docker 경로를 환경 변수에 등록할지 설정

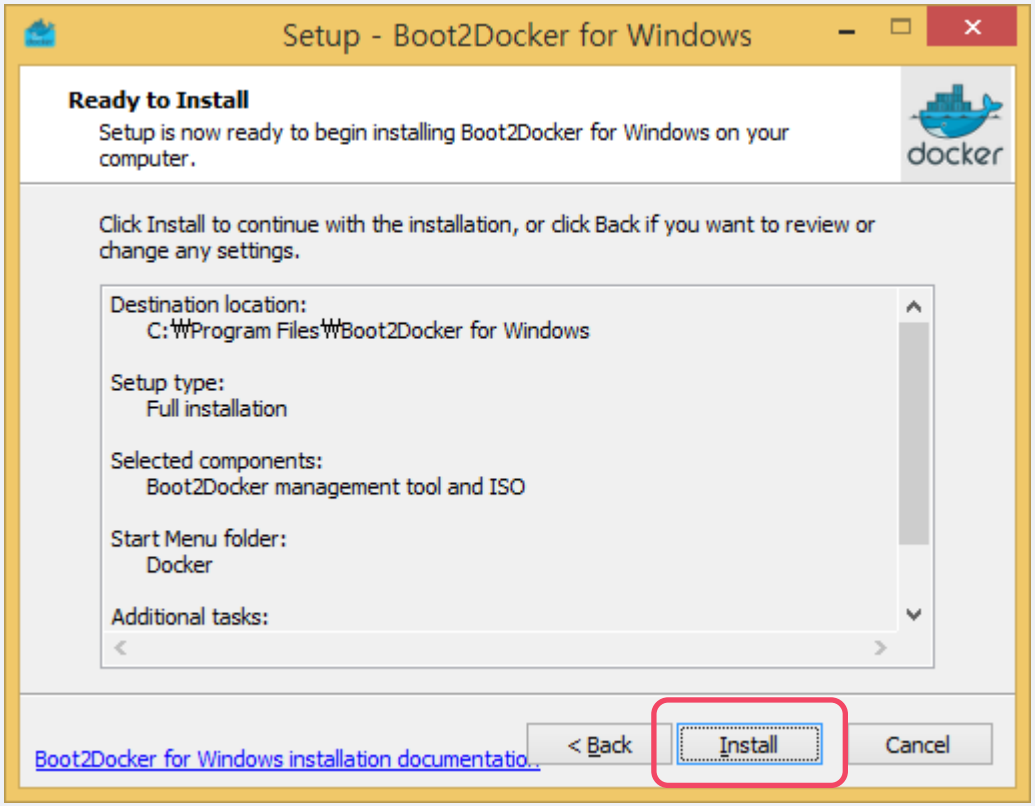
✓ 기본값 그대로 사용하고, Next 버튼을 클릭



Windows

도커 설치하기

✓ Install 버튼을 클릭하여 Boot2Docker를 설치



Windows

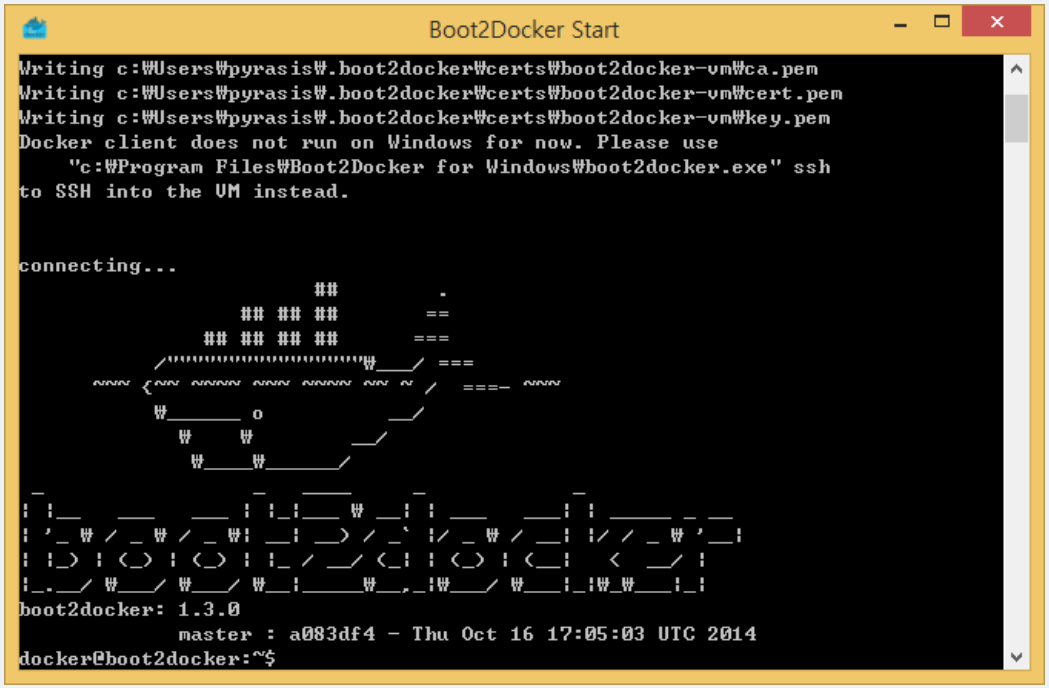
도커 설치하기

✓ 설치가 완료되면 Finish 버튼을 클릭

Windows

도커 설치하기

- ✓ 설치가 끝난 뒤 바탕화면에 생성된 Boot2Docker Start 아이콘을 더블 클릭. 만약 실행 파일 선택 창이 나오면 Git Bash의 sh.exe를 선택
- ✓ 명령 프롬프트 창이 표시된 뒤 잠시 기다리면 자동으로 boot2docker.iso를 이용하여 가상 머신이 생성되고, 가상 머신에 접속됨. 중간에 VirtualBox 관리가 권한 요청 창이 나오면 예(Y) 버튼을 클릭



도커 사용해보기

도커 명령

도커 사용해보기

docker + 〈명령〉

✓도커의 명령은 docker 〈명령〉 형식

✓예) docker run, docker push

✓항상 root 권한으로 실행

도커 사용해보기

앞으로 나오는 모든 명령을 차례대로
입력하면서 실습을 진행합니다.

search 명령으로 이미지 검색하기

도커 사용해보기

docker search <이미지 이름>

✓ Docker Hub에서 이미지 검색해보기

```
$ sudo docker search ubuntu
NAME                              DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
ubuntu                            Official Ubuntu base image                      383
stackbrew/ubuntu                  Official Ubuntu base image                      40
crashsystems/gitlab-docker        A trusted, regularly updated build of GitL...  19                   [OK]
dockerfile/ubuntu                 Trusted Ubuntu (http://www.ubuntu.com/) Bu...   15                   [OK]
ubuntu-upstart                    Upstart is an event-based replacement for ...   7
cmfatih/phantomjs                 PhantomJS [ phantomjs 1.9.7, casperjs 1.1....   5                    [OK]
dockerfile/ubuntu-desktop         Trusted Ubuntu Desktop (LXDE) (http://lxde...   5                    [OK]
lukasz/docker-scala               Dockerfile for installing Scala 2.10.3, Ja...   5                    [OK]
litaio/ruby                       Ubuntu 14.04 with Ruby 2.1.2 compiled from...   5                    [OK]
```

✓ 보통 ubuntu, centos, redis 등 OS나 프로그램 이름을 가진 이미지가 공식 이미지

✓ 나머지는 사용자들이 만들어서 공개한 이미지

search 명령으로 이미지 검색하기

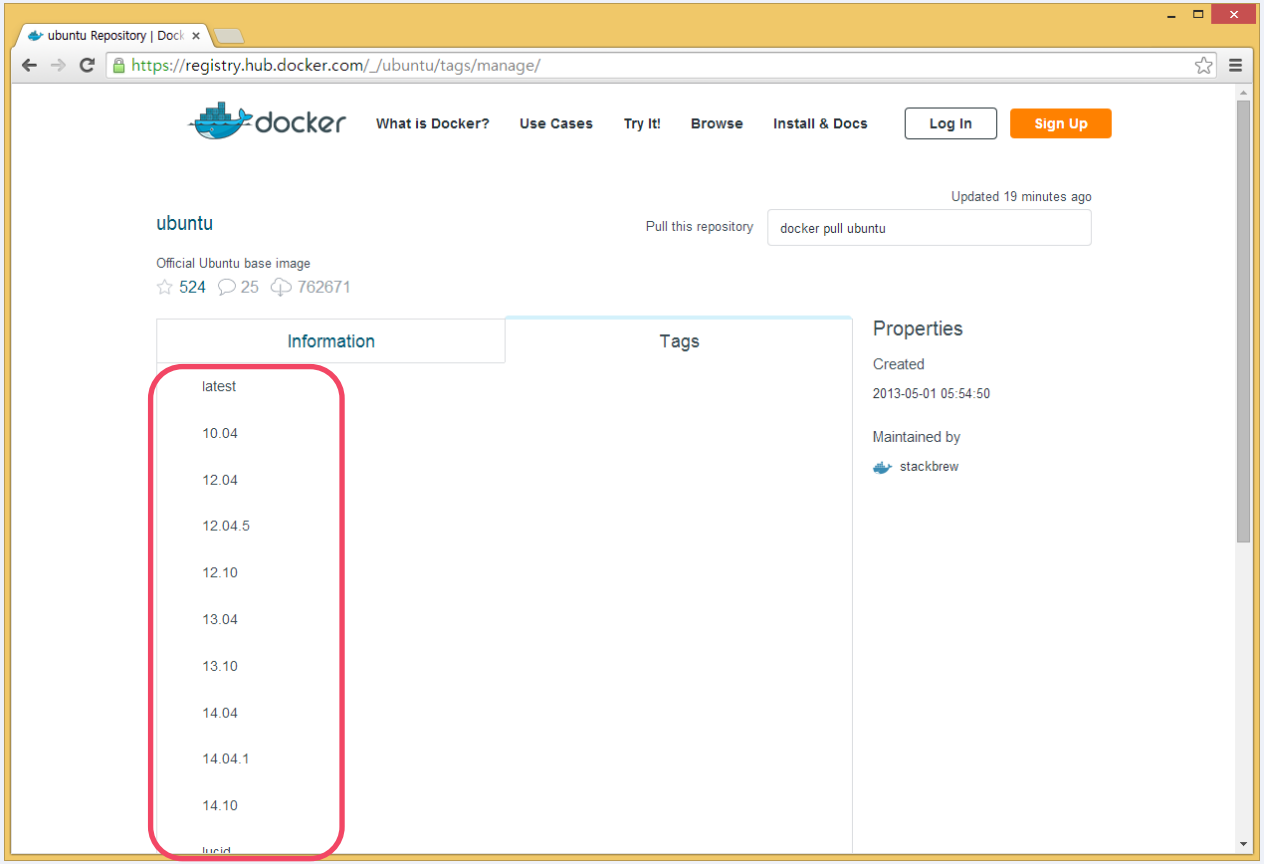
도커 사용해보기

- ✓ 도커는 Docker Hub(https://registry.hub.docker.com)를 통해 이미지를 공유하는 생태계가 구축되어 있음
- ✓ 유명 리눅스 배포판과 오픈 소스 프로젝트(Redis, Nginx 등)의 이미지를 모두 Docker Hub에서 구할 수 있음
- ✓ 이미지와 관련된 명령은 기본적으로 Docker Hub를 이용하도록 설정되어 있음

search 명령으로 이미지 검색하기

도커 사용해보기

✓ Docker Hub에서 이미지를 검색한 뒤 해당 이미지의 Tags 탭에서 이미지의 버전을 볼 수 있음



sudo 입력하지 않기

도커 사용해보기

✓ sudo를 매번 입력하지 않으려면?

처음부터 root 계정으로 로그인하거나 sudo su 명령을 사용하여 root 계정으로 전환

```
$ sudo su
#
```

또는, 현재 계정을 docker 그룹에 포함(docker 그룹은 root 권한과 동일하므로 꼭 필요한 계정만 포함)

```
$ sudo usermod -aG docker ${USER}
$ sudo service docker restart
```

현재 계정에서 로그아웃한 뒤 다시 로그인

pull 명령으로 이미지 받기

도커 사용해보기

docker pull <이미지 이름>:<태그>

✓ Docker Hub에서 우분투 이미지 받아보기

```
$ sudo docker pull ubuntu:latest
```

✓ 이미지 이름 뒤에 latest를 설정하면 최신 버전을 받음

✓ ubuntu:14.04, ubuntu:12.10처럼 태그를 지정할 수 있음

✓ 이미지 이름에서 pyrasis/ubuntu처럼 / 앞에 사용자명을 지정하면 해당 사용자가 올린 이미지를 받음(공식 이미지는 사용자명이 붙지 않음)

✓ 호스트에 설치된 리눅스 배포판과 도커 이미지의 배포판 종류는 달라도 됨. 즉 centOS에서 우분투 컨테이너를 실행할 수 있음

images 명령으로 이미지 목록 출력하기

도커 사용해보기

docker images

✓ 모든 이미지를 출력해보기

```
$ sudo docker images
REPOSITORY          TAG         IMAGE ID        CREATED                  V IRTUAL SIZE
ubuntu              latest      e54ca5efa2e9    Less than a second ago   276.1 MB
```

docker images ubuntu처럼 이미지 이름을 설정하면 이름은 같지만 태그가 다른 이미지가 출력됨

run 명령으로 컨테이너 생성하기

도커 사용해보기

docker run <옵션> <이미지 이름> <실행할 파일>

✓ 이미지를 컨테이너로 생성한 뒤 Bash 셸 실행해보기

```
$ sudo docker run -i -t --name hello ubuntu /bin/bash
```

ubuntu 이미지를 컨테이너로 생성한 뒤 ubuntu 이미지 안의 /bin/bash를 실행

✓ -i(interactive), -t(Pseudo-tty) 옵션을 사용하면 실행된 Bash 셸에 입력 및 출력 가능

✓ --name 옵션으로 컨테이너에 이름을 지정할 수 있음. 이름을 지정하지 않으면 도커가 자동으로 이름을 생성하여 지정

run 명령으로 컨테이너 생성하기

도커 사용해보기

cd, ls 명령으로 컨테이너 내부를 둘러본 뒤 exit를 입력하여 Bash 셸에서 빠져나오기

✓ 우분투 이미지에서 /bin/bash 실행 파일을 실행했기 때문에 여기서 빠져나오면 컨테이너가 정지(stop)됨

run 명령으로 컨테이너 생성하기

도커 사용해보기

참고

centOS에서 다음과 같은 에러가 발생한다면

```
unable to remount sys readonly: unable to mount sys as readonly max retries reached
```

/etc/sysconfig/docker 파일에서 다음과 같이 --exec-driver=lxc를 추가

```
# /etc/sysconfig/docker
#
# Other arguments to pass to the docker daemon process
# These will be parsed by the sysv initscript and appended
# to the arguments list passed to docker -d

other_args="--selinux-enabled --exec-driver=lxc"
```

Docker 서비스 재시작

```
$ sudo service docker restart
```

PS 명령으로 컨테이너 목록 확인하기

도커 사용해보기

docker ps

✓ 모든 컨테이너 목록을 출력하기

```
$ sudo docker ps -a
CONTAINER ID  IMAGE          COMMAND    CREATED              STATUS                          PORTS   NAMES
6338ce52d07c  ubuntu:latest  /bin/bash  4 seconds ago Exited (0) Less than a second ago          hello
```

현재 컨테이너는 정지된 상태. 앞에서 컨테이너를 생성할 때 이름을 hello로 지정했으므로 컨테이너 목록에서도 hello로 표시

✓ -a 옵션을 사용하면 정지된 컨테이너까지 모두 출력됨

✓ 옵션을 사용하지 않으면 실행되고 있는 컨테이너만 출력됨

start 명령으로 컨테이너 시작하기

도커 사용해보기

docker start <컨테이너 이름>

✓ 방금 정지한 컨테이너를 다시 시작하기

```
$ sudo docker start hello
```

컨테이너 이름 대신 컨테이너 ID를 사용해도 됨

✓ 컨테이너를 시작했으면 컨테이너 목록 출력해보기

```
$ sudo docker ps
CONTAINER ID  IMAGE          COMMAND    CREATED             STATUS                    PORTS   NAMES
6338ce52d07c  ubuntu:latest  /bin/bash  15 minutes ago      Up 3 seconds                      hello
```

hello 컨테이너가 시작되었음

restart 명령으로 컨테이너 재시작하기

도커 사용해보기

docker restart <컨테이너 이름>

✓ OS 재부팅처럼 컨테이너를 다시 시작해보기

```
$ sudo docker restart hello
```

컨테이너 이름 대신 컨테이너 ID를 사용해도 됨

attach 명령으로 컨테이너에 접속하기

도커 사용해보기

docker attach <컨테이너 이름>

✓ 방금 시작한 컨테이너에 접속해보기

✓ 명령을 실행한 뒤 엔터를 한 번 더 입력하면 Bash 셸이 표시됨

```
$ sudo docker attach hello
root@6338ce52d07c:/#
```

✓ 컨테이너 이름 대신 컨테이너 ID를 사용해도 됨

✓ 여기서는 /bin/bash를 실행했기 때문에 명령을 자유롭게 입력할 수 있음. 단 DB나 서버 애플리케이션을 실행하면 입력은 할 수 없고 출력만 보게됨

✓ Bash 셸에서 exit 또는 ctrl+D를 입력하면 컨테이너가 정지됨

여기서는 ctrl+P, ctrl+Q를 차례대로 입력하여 컨테이너를 정지하지 않고, 빠져 나오기

exec 명령으로 외부에서 컨테이너 안의 명령 실행하기

도커 사용해보기

docker exec <컨테이너 이름> <명령> <매개 변수>

✓ 현재 컨테이너가 /bin/bash로 실행된 상태

✓ /bin/bash를 통하지 않고 외부에서 컨테이너 안의 명령 실행해보기

```
$ sudo docker exec hello echo "Hello World"
Hello World
```

✓ 컨테이너 이름 대신 컨테이너 ID를 사용해도 됨

✓ 컨테이너가 실행되고 있는 상태에서만 사용할 수 있으며 정지된 상태에서는 사용할 수 없음

컨테이너 안의 echo 명령을 실행하고 매개 변수로 "Hello World"를 지정했기 때문에 Hello World가 출력됨

✓ docker exec 명령은 이미 실행된 컨테이너에 apt-get, yum 명령으로 패키지를 설치하거나 각종 데몬을 실행할 때 활용

stop 명령으로 컨테이너 정지하기

도커 사용해보기

docker stop <컨테이너 이름>

✓ 먼저 실행된 컨테이너 목록 출력하기

```
$ sudo docker ps
CONTAINER ID  IMAGE          COMMAND    CREATED             STATUS                    PORTS   NAMES
6338ce52d07c  ubuntu:latest  /bin/bash  51 minutes ago      Up 2 minutes                      hello
```

✓ 컨테이너 정지해보기(컨테이너 이름 대신 컨테이너 ID도 사용해도 됨)

```
$ sudo docker stop hello
```

✓ 실행된 컨테이너 목록을 출력해보기

```
$ sudo docker ps
CONTAINER ID  IMAGE          COMMAND    CREATED             STATUS                    PORTS   NAMES
```

hello 컨테이너를 정지했기 때문에 아무것도 나오지 않음

rm 명령으로 컨테이너 삭제하기

도커 사용해보기

docker rm <컨테이너 이름>

✓ 생성된 컨테이너를 삭제해보기

```
$ sudo docker rm hello
```

컨테이너 이름 대신 컨테이너 ID를 사용해도 됨

✓ 모든 컨테이너 목록을 출력해보기

```
$ sudo docker ps -a
CONTAINER ID  IMAGE         COMMAND    CREATED             STATUS                      PORTS    NAMES
```

hello 컨테이너를 삭제했기 때문에 아무것도 나오지 않음

rmi 명령으로 이미지 삭제하기

도커 사용해보기

docker rmi <이미지 이름>:<태그>

✓ 이미지 삭제해보기

```
$ sudo docker rmi ubuntu:latest
```

✓ 이미지 이름 대신 이미지 ID를 사용해도 됨

✓ docker rmi ubuntu처럼 이미지 이름만 지정하면 태그는 다르지만 ubuntu 이름을 가진 모든 이미지가 삭제됨

✓ 이미지 목록을 출력해보기

```
$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
```

ubuntu 이미지를 삭제했기 때문에 아무것도 나오지 않음

도커 이미지 생성하기

Dockerfile

✓ 도커 이미지 설정 파일

✓ Dockerfile에 설정된 대로 이미지를 생성하게 됨

Dockerfile 작성하기

도커 이미지 생성하기

✓ 먼저 example 디렉터리를 생성한 뒤 example 디렉터리로 이동하기

```
~$ mkdir example
~$ cd example
```

Dockerfile 작성하기

도커 이미지 생성하기

✓ 다음 내용을 example 디렉터리 아래에 Dockerfile로 저장하기

✓ 우분투 14.04를 기반으로 nginx 서버를 설치한 도커 이미지를 생성하는 예제

example/Dockerfile

```
FROM ubuntu:14.04        # 어떤 이미지를 기반으로 할지 설정. <이미지 이름>:<태그> 형식
MAINTAINER Foo Bar foo@bar.com   # 메인테이너 정보

RUN apt-get update     # RUN으로 셸 스크립트 혹은 명령 실행
RUN apt-get install -y nginx                                  # 이미지 생성 중에는 사용자 입력을 받을 수 없으므로
RUN echo "\ndaemon off;" >> /etc/nginx/nginx.conf             apt-get install에서 -y 옵션 사용
RUN chown -R www-data:www-data /var/lib/nginx

VOLUME ["/data", "/etc/nginx/site-enabled", "/var/log/nginx"]  # 호스트와 공유할 디렉터리 목록

WORKDIR /etc/nginx   # 다음 CMD에서 설정한 실행 파일이 실행될 디렉터리

CMD ["nginx"]   # 컨테이너가 시작되었을 때 실행할 실행 파일 또는 스크립트

EXPOSE 80
EXPOSE 443     # 호스트와 연결할 포트 번호
```

https://github.com/pyrasis/dockerbook/blob/master/Chapter04/Dockerfile

build 명령으로 이미지 생성하기

도커 이미지 생성하기

docker build <옵션> <Dockerfile 경로>

✓ 앞에서 작성한 Dockerfile로 이미지 생성해보기

✓ Dockerfile이 저장된 example 디렉터리에서 명령 실행

```
~/example$ sudo docker build --tag hello:0.1 .
```

✓ --tag 옵션으로 이미지 이름과 태그를 설정할 수 있음

✓ 이미지 이름만 설정하면 태그는 latest로 설정됨

✓ 잠시 기다리면 이미지가 생성됨. 이미지 목록 출력해보기

```
$ sudo docker images
REPOSITORY      TAG         IMAGE ID        CREATED                  VIRTUAL SIZE
ubuntu          14.04       e54ca5efa2e9    Less than a second ago   276.1 MB
ubuntu          latest      e54ca5efa2e9    Less than a second ago   276.1 MB
hello           0.1         2031ee0736e8    9 minutes ago            298.4 MB
```

hello:0.1 이미지가 생성됨

build 명령으로 이미지 생성하기

도커 이미지 생성하기

✓ 앞에서 생성한 이미지를 실행해보기

```
$ sudo docker run --name hello-nginx -d -p 80:80 -v /root/data:/data hello:0.1
```

✓ -d 옵션은 컨테이너를 백그라운드로 실행

✓ -p 80:80 옵션으로 호스트의 80번 포트와 컨테이너의 80번 포트를 연결하고 외부에 노출

✓ -v /root/data:/data 옵션으로 호스트의 /root/data 디렉터리를 컨테이너의 /data 디렉터리에 연결

build 명령으로 이미지 생성하기

도커 이미지 생성하기

✓ 실행된 컨테이너 목록을 출력해보기

```
$ sudo docker ps
CONTAINER ID  IMAGE      COMMAND  CREATED         STATUS         PORTS                         NAMES
3c06a0bebab6  hello:0.1  nginx    10 minutes ago  Up 10 minutes  443/tcp, 0.0.0.0:80->80/tcp   hello-nginx
```

hello-nginx 컨테이너가 실행되었음

웹 브라우저를 실행하고 http://<호스트 IP>:80으로 접속하면

Welcome to nginx! 페이지가 표시됨

build 명령으로 이미지 생성하기

도커 이미지 생성하기

✓ 만약 Boot2Docker를 사용한다면?

✓ Boot2Docker는 가상 머신 안에 도커를 실행한 것이므로 호스트 IP로는 nginx에 바로 접속할 수 없다.

✓ boot2docker ip 명령으로 가상 머신의 IP 주소 알아내기

✓ Windows에서는 Git Bash에서 다음 명령 실행

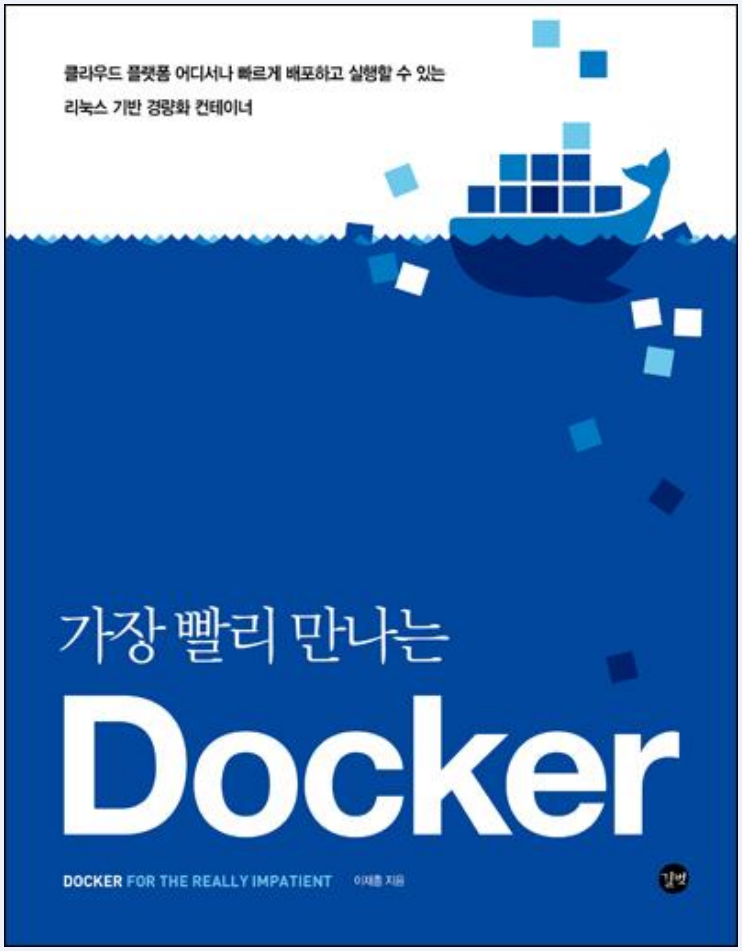
```
$ boot2docker ip

The VM's Host only interface IP address is: 192.168.59.103
```

Boot2Docker 가상 머신의 IP 주소가 출력됨

웹 브라우저를 실행하고 http://<Boot2Docker VM의 IP>:80으로 접속하면

Welcome to nginx! 페이지가 표시됨

더 자세한 내용은...



더 자세한 내용은 가장 빨리 만나는 도커를 참조해주세요. 책 내용은 모두 웹사이트에 공개되어 있습니다.

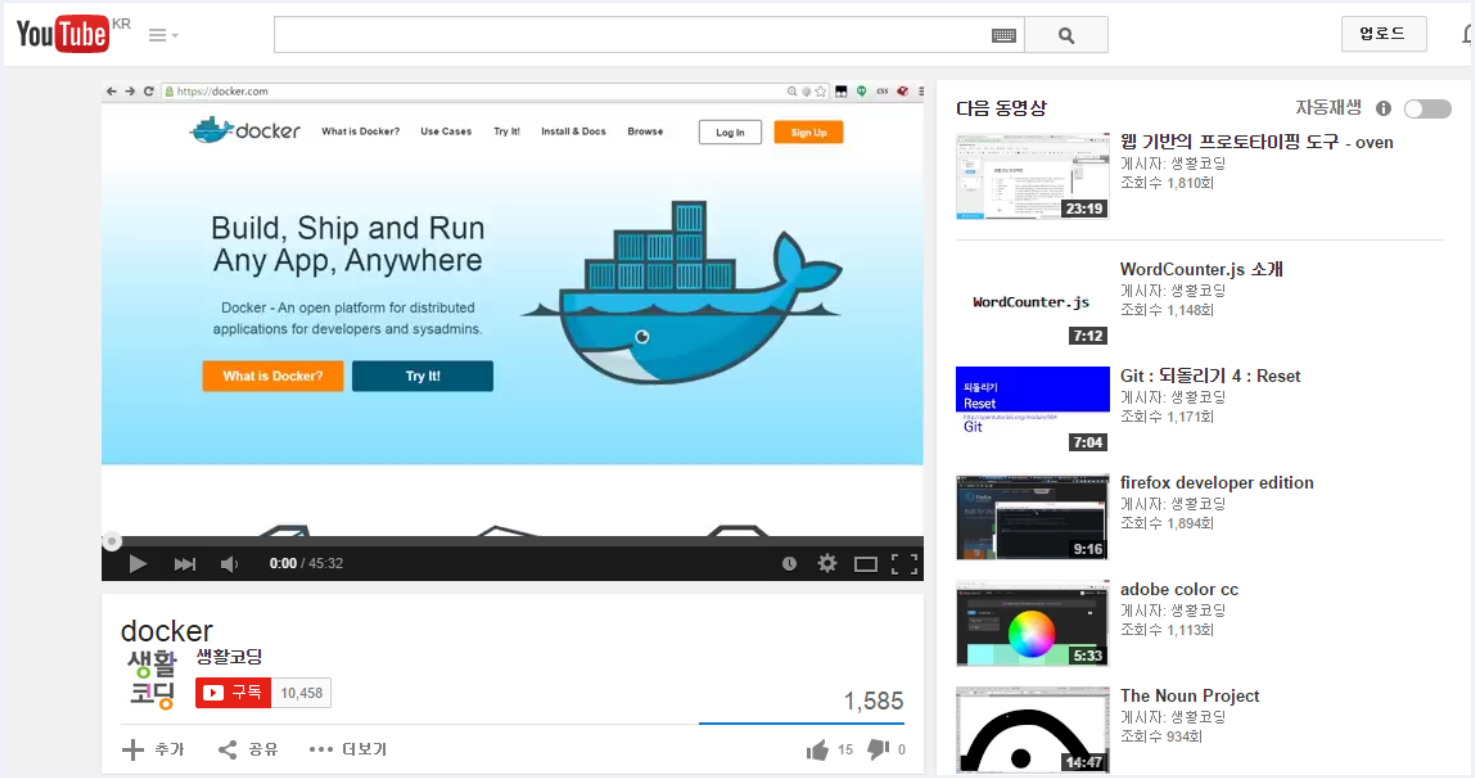
http://www.pyrasis.com/private/2014/11/30/publish-docker-for-the-really-impatient-book

- ✓ Docker 개인 저장소 구축하기
- ✓ Dockerfile 상세 설명
- ✓ Docker로 애플리케이션 배포하기
- ✓ Docker 모니터링하기
- ✓ Amazon Web Services에서 Docker 사용하기
- ✓ Google Cloud Platform에서 Docker 사용하기
- ✓ Microsoft Azure에서 Docker 사용하기
- ✓ Docker Hub 사용하기
- ✓ Docker Remote API 사용하기
- ✓ CoreOS 사용하기
- ✓ Docker로 워드프레스 블로그 구축하기(MySQL)
- ✓ Docker로 Ruby on Rails 애플리케이션 구축하기(MySQL, PostgreSQL)
- ✓ Docker로 Django 애플리케이션 구축하기(Oracle, MySQL, PostgreSQL)
- ✓ Docker 활용 시나리오
- ✓ Docker 명령어 및 옵션 목록

YouTube 동영상 강의

생활코딩 이고잉님과 함께하는 저의 동영상 강의입니다.

https://www.youtube.com/watch?v=Bhzz9E3xuXY



감사합니다