



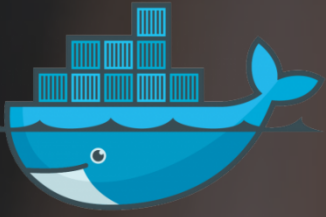
# Docker Hands-on Lab

*Babel Search*



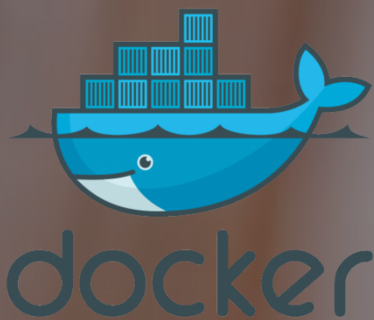
# Docker Hands-on Lab

*Babel Search*



# Getting Started with Docker

# Contents



Docker 알아보기 WHAT?



Docker 설치하기 HOW?



Docker 시작하기 HELLO!





# Docker 알아보기 WHAT?

## Virtual Machine

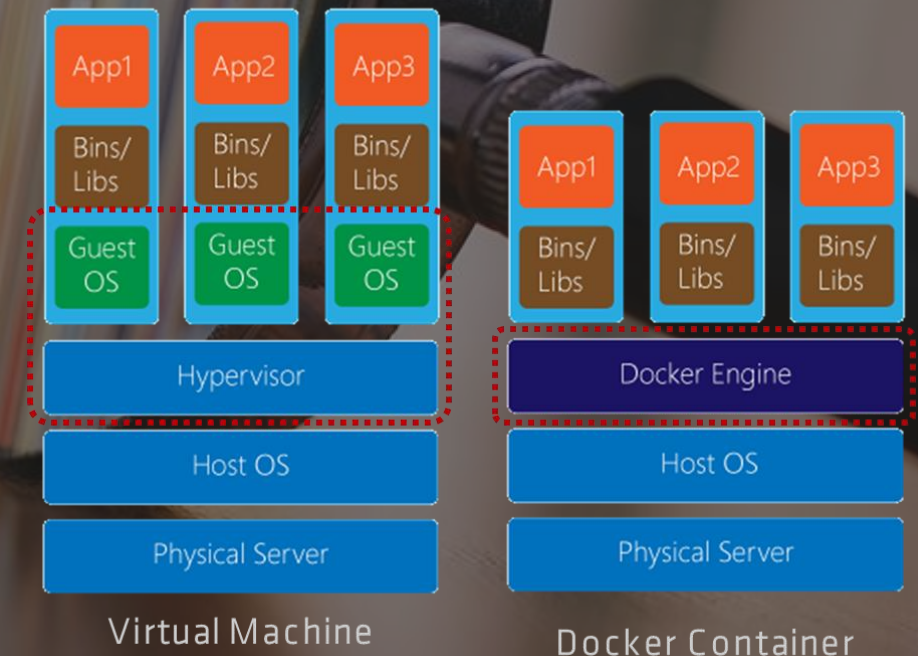
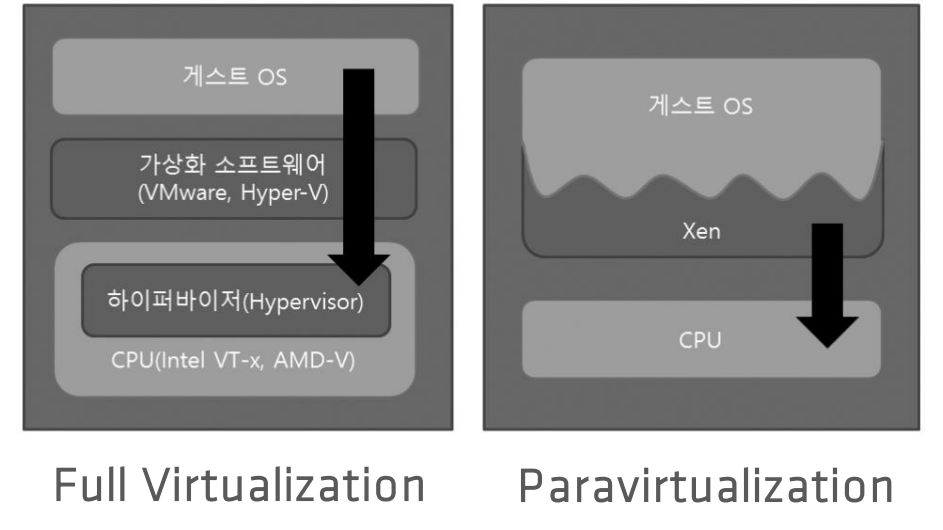
하드웨어 가상화로 인해 성능 이슈  
성능 개선을 위한 Paravirtualization 기술  
Guest OS의 리소스 점유  
가상화 OSS : ~~생성~~ ~~실행~~ ~~배포~~ ~~관리~~

## 관리/배포 상용 제품

VMWare vCenter, MS System Center

## Container (Docker)

OS 자원 (시스템 콜)은 호스트OS와 공유  
메모리 접근, 파일시스템, 네트워크 성능 개선  
이미지 경량화, 리눅스, 다양한 관리도구 지원





# Docker 알아보기 WHAT?

## Docker의 성능 비교 측정

호스트 환경 : Ubuntu 14.04

도커 환경 : Docker 1.1.2 (Ubuntu 14.04)

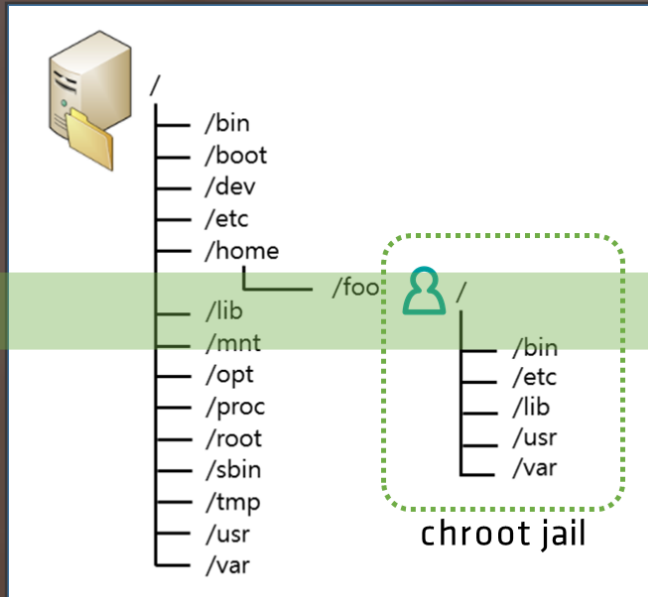
Platform	성능측정도구	호스트	Docker
CPU	sysbench	1	0.9945
Memory Write	sysbench	1	0.9826
Memory Read	sysbench	1	1.0025
Disk I/O	dd	1	0.9811
Network	iperf	1	0.9626



# Docker 알아보기 WHAT?

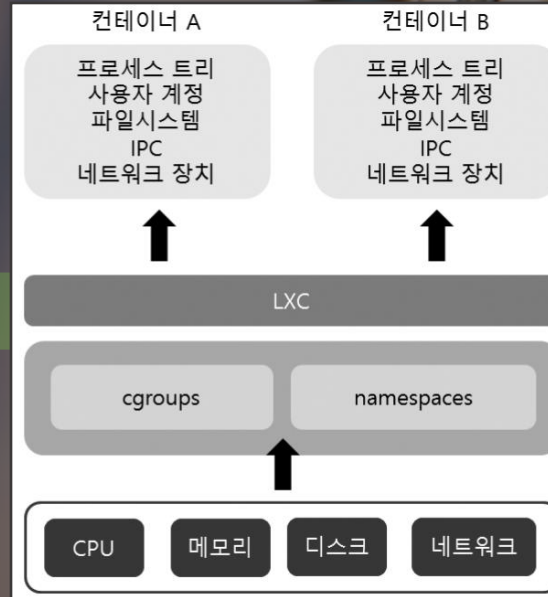
## 리눅스 컨테이너의 역사

### chroot



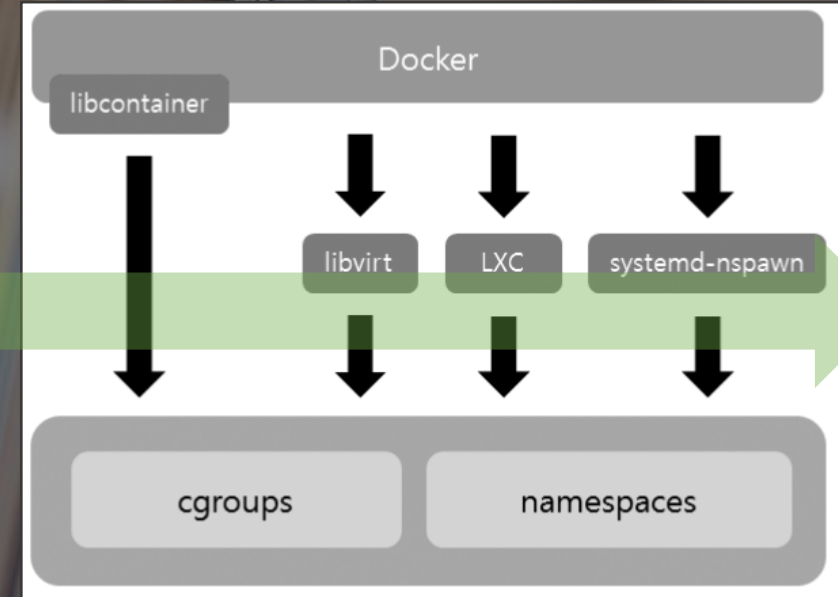
외부 디렉토리 경로와 격리  
서버 정보 유출 방지 최소화 목적  
설정이 복잡하고, 제약이 많음

### LXC(Linux Container)



LXC : 커널에서 제공하는 cgroups와 namespace로 격리된 컨테이너 구성  
cgroups : CPU, 메모리, 디스크, 네트워크 자원을 가상공간에 할당하는 기능 제공  
namespaces : 프로세스 트리, 사용자 계정, 파일시스템, IPC 등을 호스트와 격리  
도커는 초기에 LXC를 실행 드라이버(exec driver)로 채택하여 컨테이너 관리  
Docker 0.9 이후 실행 드라이버로 자체 개발한 libcontainer를 사용

### Docker Container







# Docker 알아보기 WHAT?

## Docker 1.10 and Later



### Linux Containers



liblxc

namespaces

cgroups

SELinux/AppArmor

Linux kernel



### Docker 1.10 and later



runC

runC

runC

containerd-shim

containerd-shim

containerd-shim

containerd

Docker Engine

namespaces

cgroups

SELinux/AppArmor

Linux kernel





# Docker 알아보기 WHAT?

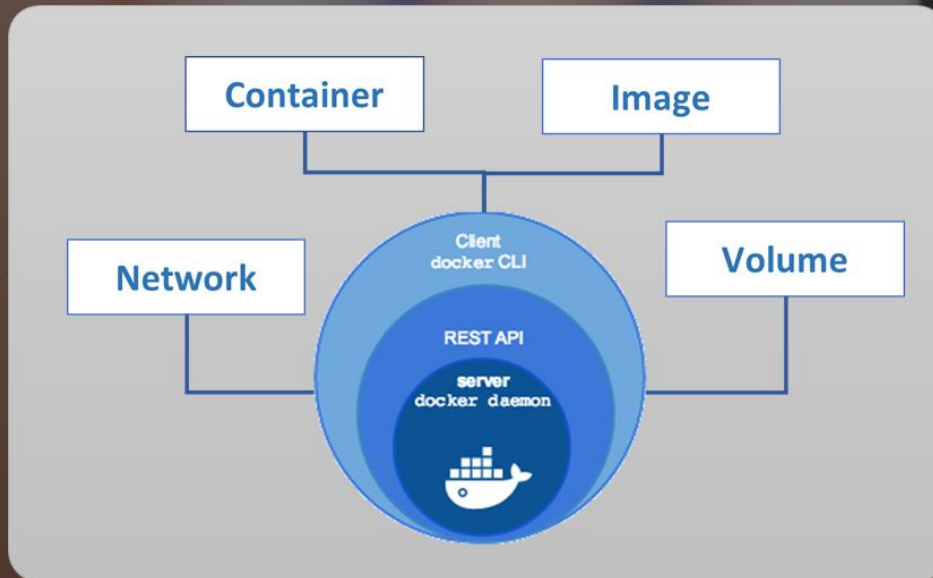
## Docker Engine (사용자 관점)

Docker CLI를 이용하여 컨테이너 관리

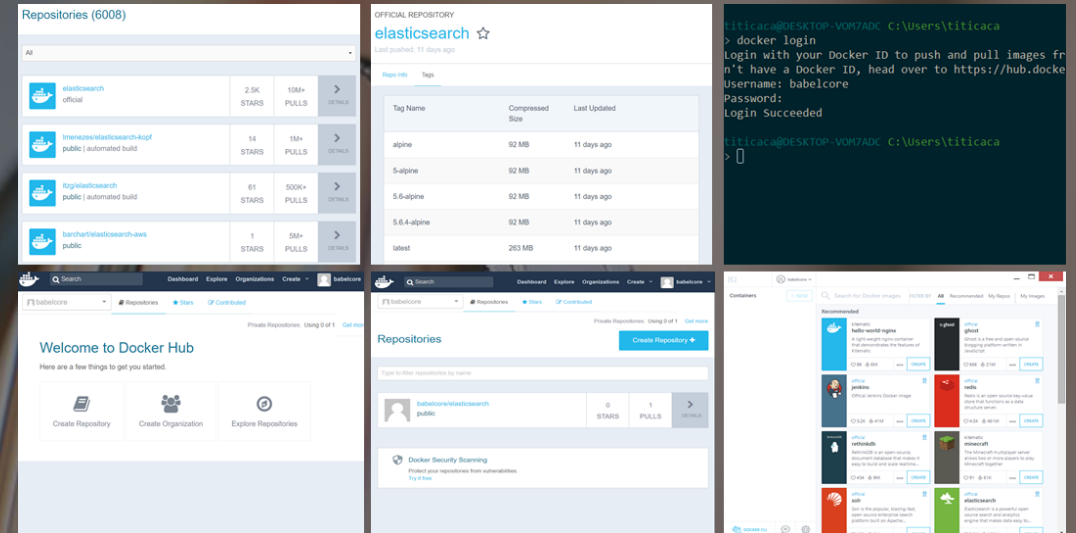
\* CLI (Command Line Interface)

## Docker Registry

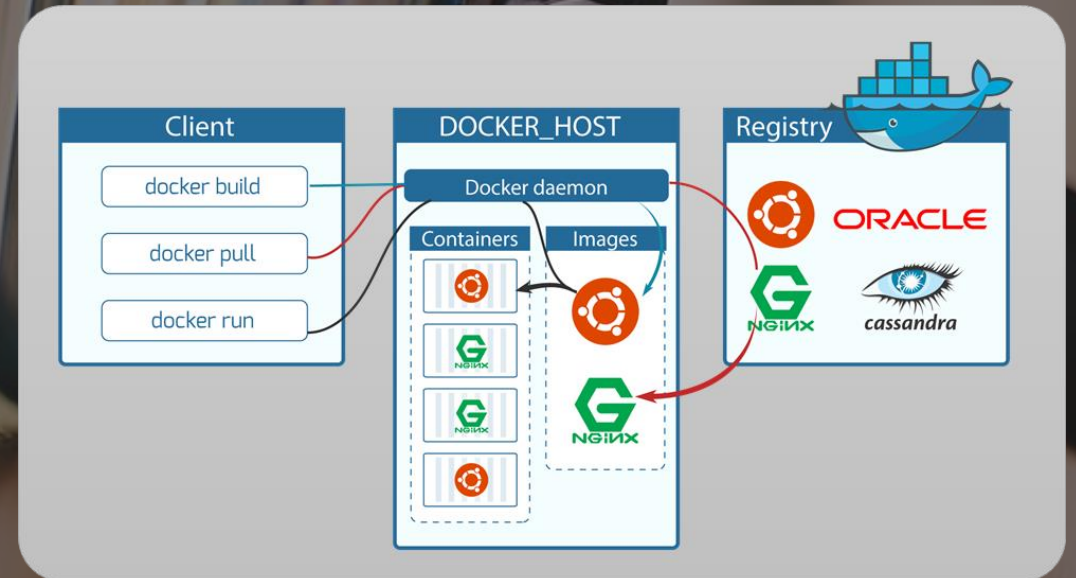
Docker 엔진과 Docker Registry 환상적 콜라보  
클라우드 서비스도 별도의 Registry 운영



Docker Engine의 구성과 역할



Docker Registry Environment



Docker Environment



# Docker 알아보기 WHAT?

## Docker Image

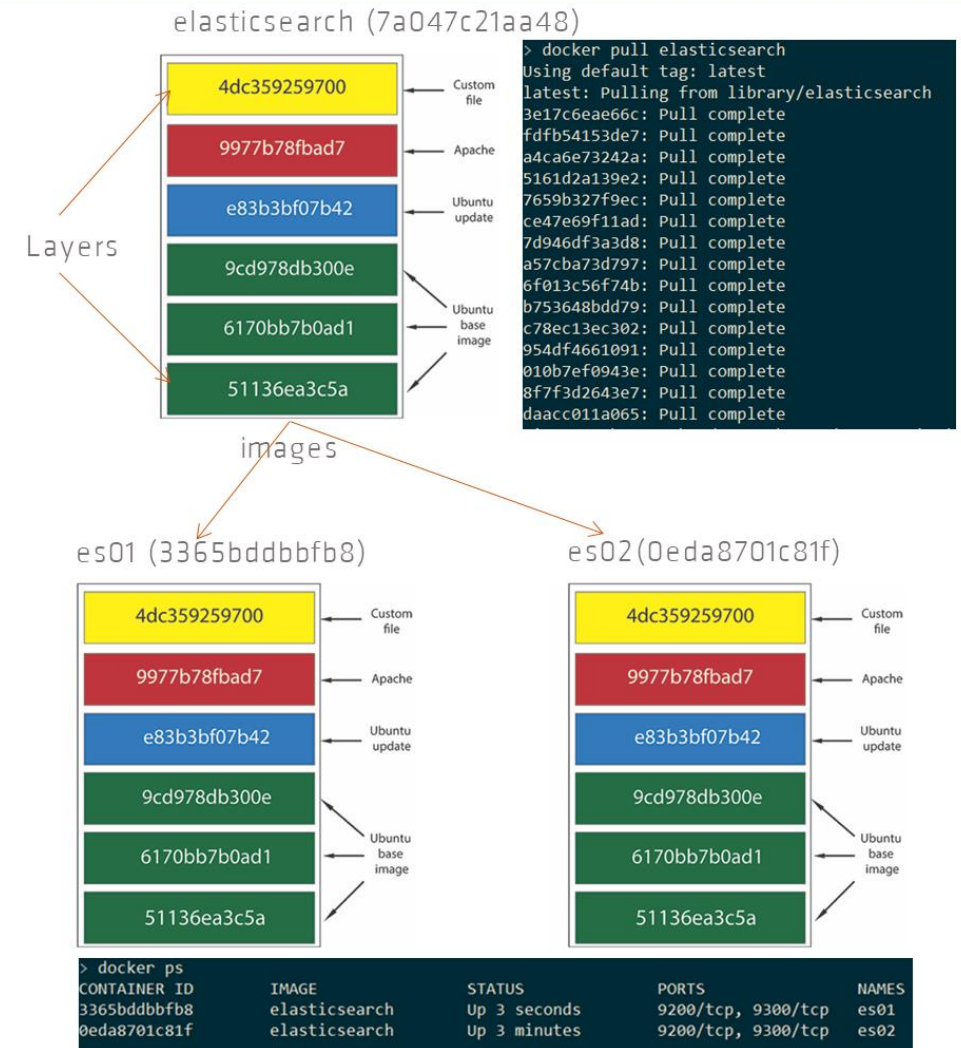
여러 개의 Layer로 구성

Layer의 종류

- Base Image
- OS Updates
- Components
- Custom Files

## Docker Container

운영체제 관점에서 Image가 실행파일이며,  
Container는 프로세스



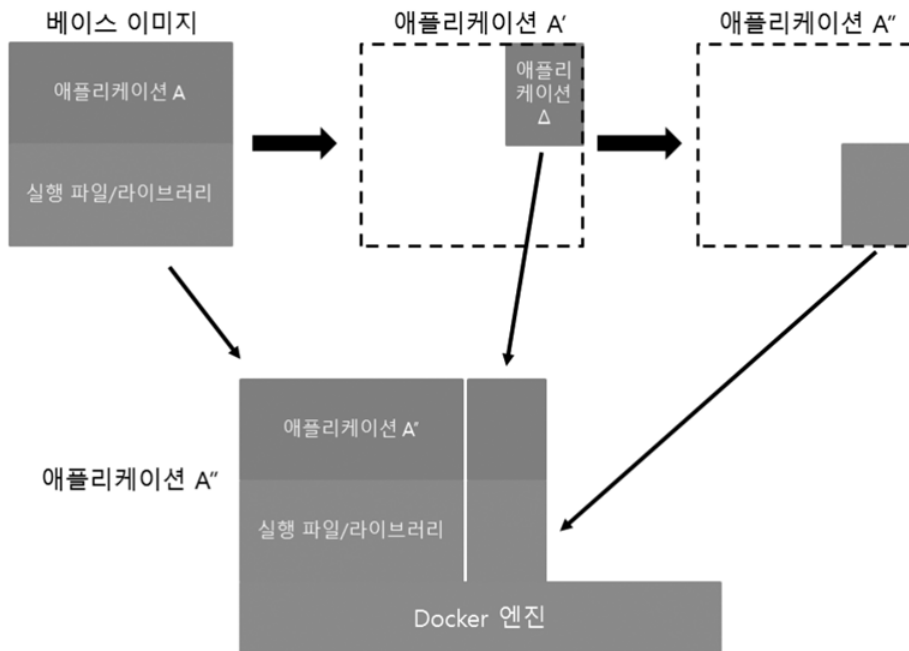
Docker Image, Layer, Container Relationship



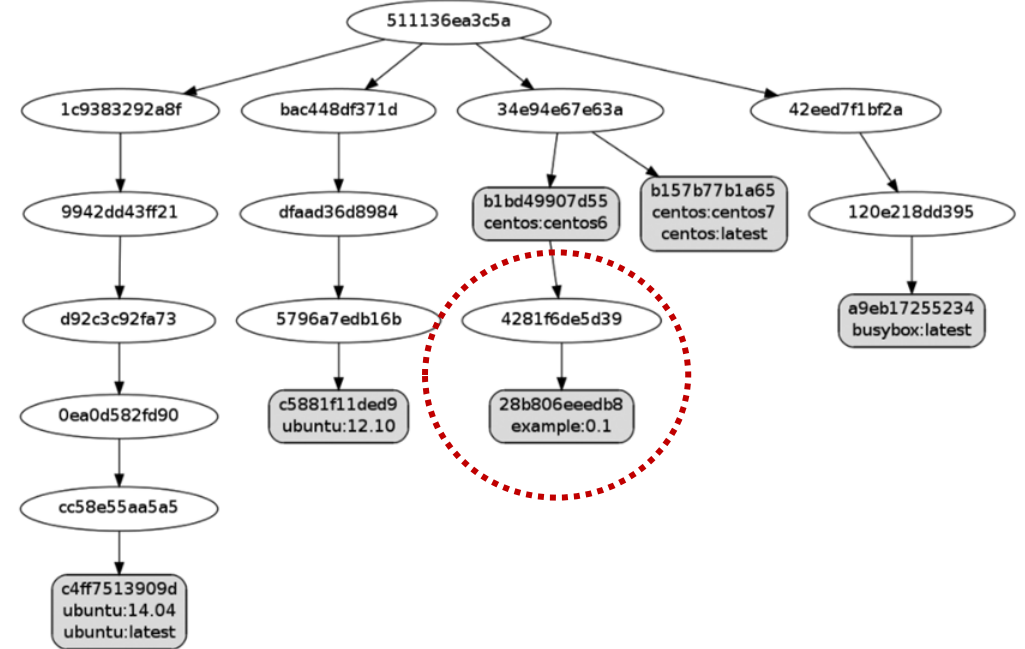
# Docker 알아보기 WHAT?

## Docker Image Dependency

컨테이너의 변경 부분을 인지하여 기존 이미지와 변경 부분을 합쳐 컨테이너 실행  
변경 부분을 합쳐서 새로운 이미지를 생성이 가능



Docker 이미지 변경점 관리



Docker Layer Dependency



# Docker 설치하기 HOW?

## Docker Editions

Community Edition(CE) : 작은 조직과 컨테이너 기반의 프로그램 개발을 위한 실험 환경 구성 목적. 빠른 패치

Enterprise Edition(EE) : 비즈니스 목적으로 구축된 프로그램을 운영하는 목적. 관리, 보안기능 강화 (유료)

Capabilities	CE	EE Basic	EE Standard	EE Advanced
Container engine and built in orchestration, networking, security	✓	✓	✓	✓
Certified infrastructure, plugins and ISV containers		✓	✓	✓
Image management			✓	✓
Container app management			✓	✓
Image security scanning				✓





# Docker 설치하기 HOW?

## Supported Platforms

### Desktop

Platform	CE x86_64	CE ARM	EE
Docker for Mac	✓		
Docker for Windows	✓		

### Cloud

Platform	CE x86_64	CE ARM	EE
Amazon Web Service	✓		✓
Microsoft Azure	✓		✓

### Server

Platform	CE x86_64	CE ARM	CE IBM Z	EE x86_64	EE IBM Z
CentOS	✓			✓	
Debian	✓	✓			
Fedora	✓				
Window Server 2016 Oracle Linux				✓	
Red Hat Enterprise Linux SUSE Linux Enterprise Server				✓	✓
Ubuntu	✓	✓	✓	✓	✓



# Docker 설치하기 HOW?

## Docker 설치하기

SKIP!

Windows : Window 10 이상은 Docker for Windows. Windows 10 이하는 Docker Toolbox.

LAB

Mac : Docker for Mac 설치

LAB

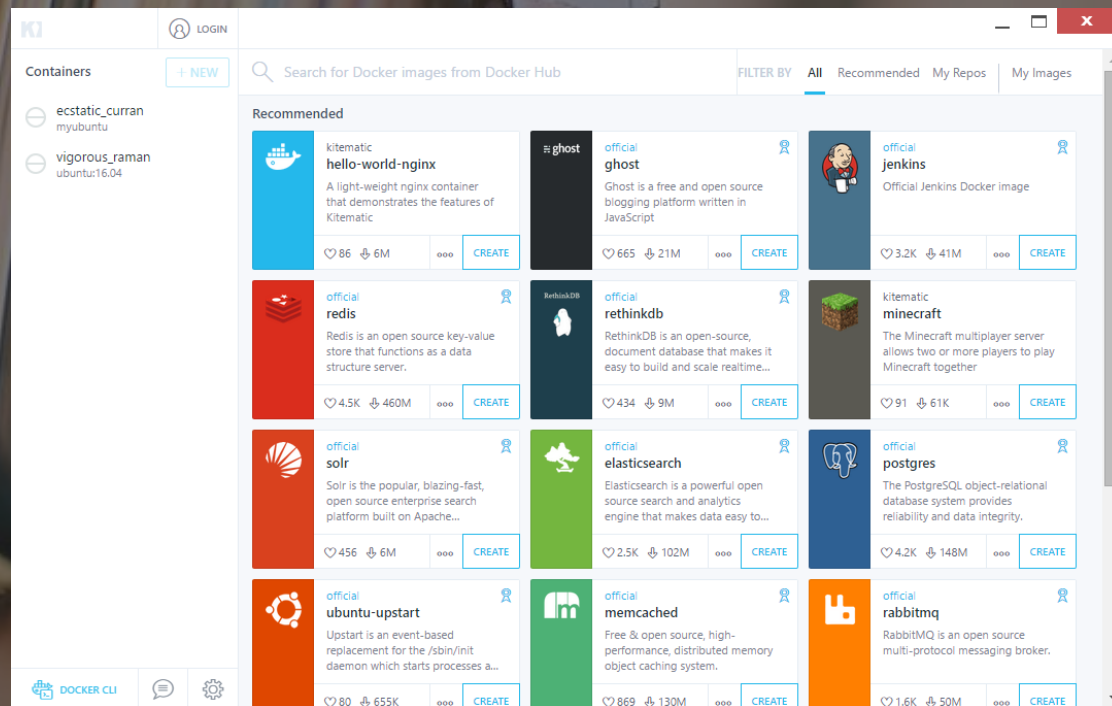
Linux : Cloud 위에 Ubuntu 서버 생성하여 실습진행

## Kitematic 설치하기

Docker UI 관리도구

Docker CLI의 많은 기능을 UI로 처리

Docker 모니터링 기능



Kitematic 예시 화면



# Docker CLI INTERFACE

## Docker CLI Basic Commands

`docker run`

`docker ps`

`docker search`

`docker images`

`docker rm`

`docker rmi`

`docker build`

`docker pull`

`docker push`

`docker commit`

