

오픈소스 SW 입문

Docker & Kubernetes

팀 J

2019125017 김종현

2021125015 김재영

2021128032 임유한

2020126102 지윤진

2023125065 홍지호

CONTENTS

Docker & Kubernetes

1 개요(탄생배경)

2 도커

3 쿠버네티스

4 커뮤니티

* 사용방법

01

개요

등장배경

DevOps

MSA

DevOps란?



문제점

- 소프트웨어 개발과 운영의 분리
- 각 팀이 지닌 다른 목표와 책임
- 소통 부족

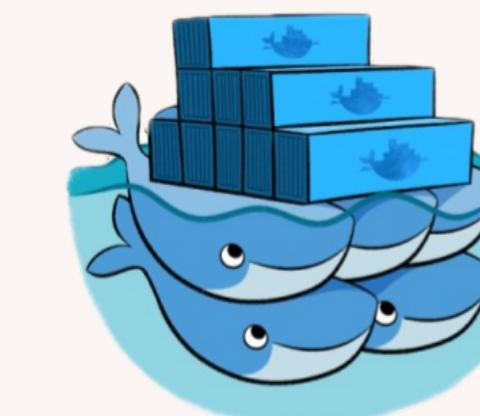
해결책

- 하나의 목표 아래 두 팀의 협력
- DevOps(Development + Operation)

등장

장점

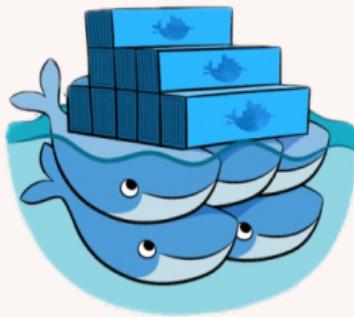
- 소프트웨어 개발과 운영 간 유기적 연동
- 소프트웨어 라이프사이클의 효율적 관리
- 문제해결 속도 및 서비스 안정성 향상
- 사용자 만족도 증대



kubernetes

DevOps 관련개념

Container



Docker

- 애플리케이션과 그 실행환경을 컨테이너에 패키징
- 언제든 동일한 환경에서 애플리케이션 실행 가능
- 개발환경과 운영환경의 차이로 발생하는 문제점 극복

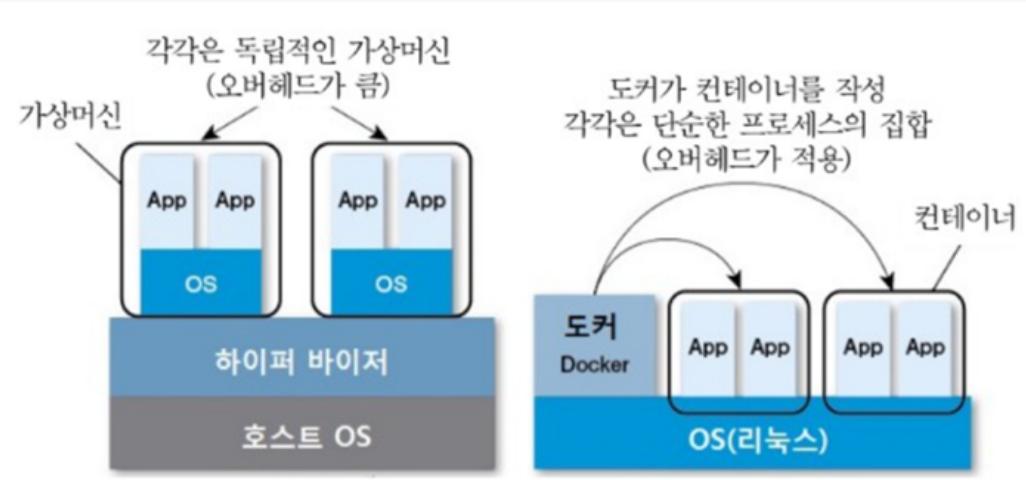
Container Orchestration



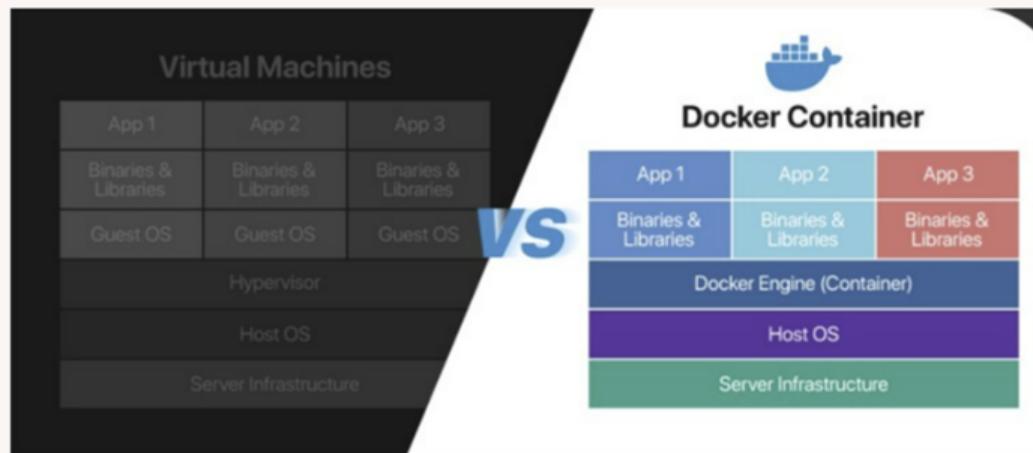
Kubernetes

- 컨테이너화된 애플리케이션의 배포, 확장, 관리 자동화 플랫폼
- 대규모의 컨테이너를 효율적으로 관리

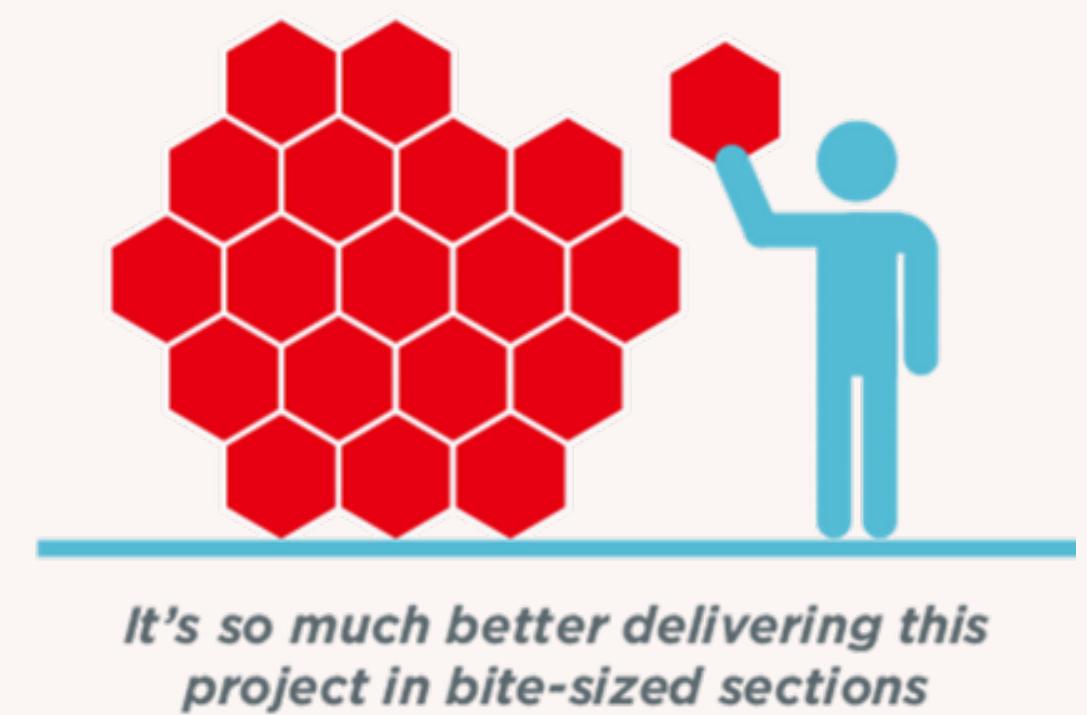
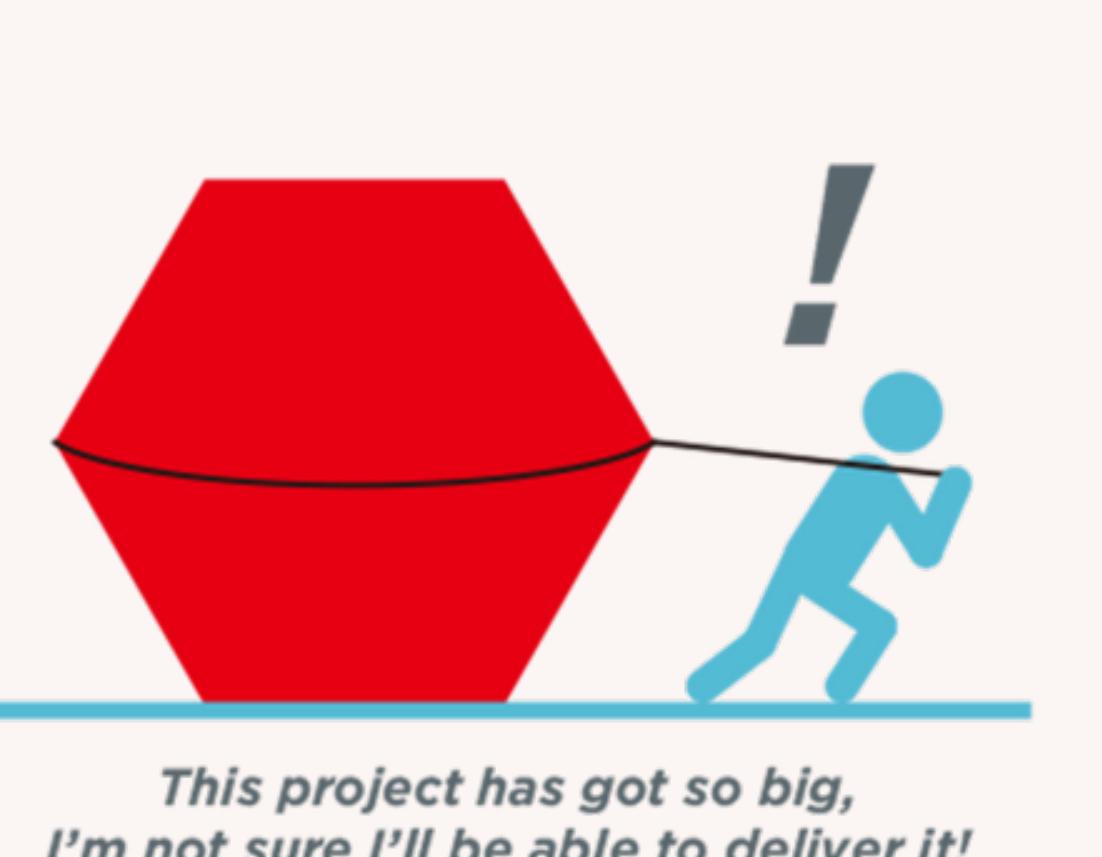
MSA



- 기존 하이퍼 바이저의 문제점과 그 문제점을 해결



- VM(가상 머신)과 Docker 비교



- 모놀리식 아키텍처와 MSA 아키텍처의 차이

02

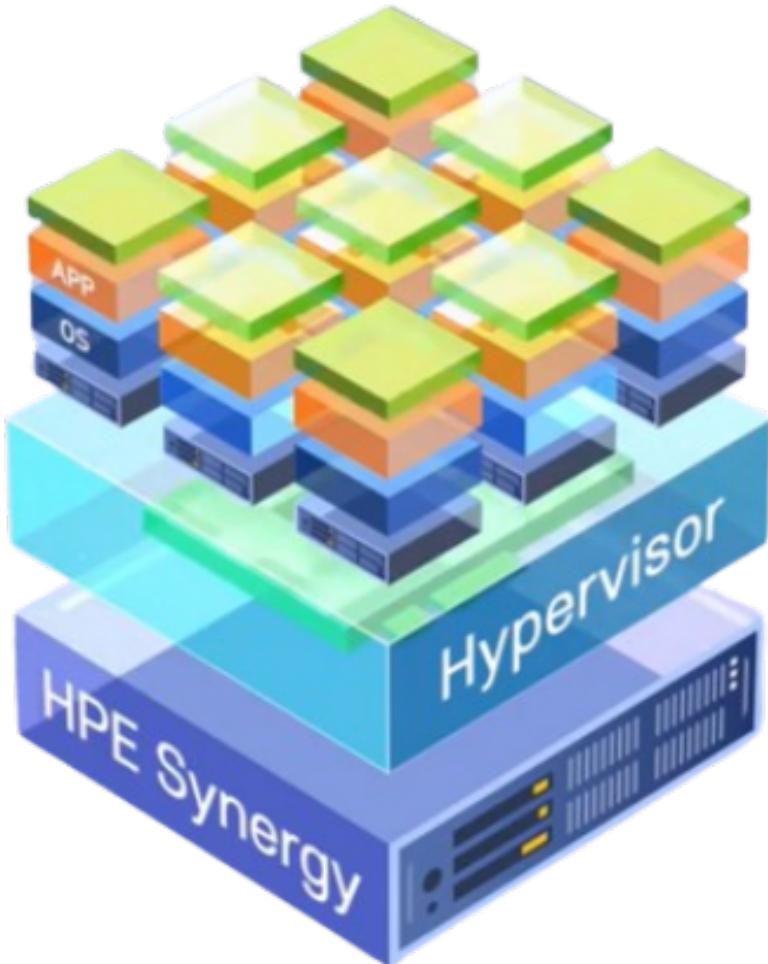
Docker

가상화

컨테이너

이미지

Docker 소개



가상화

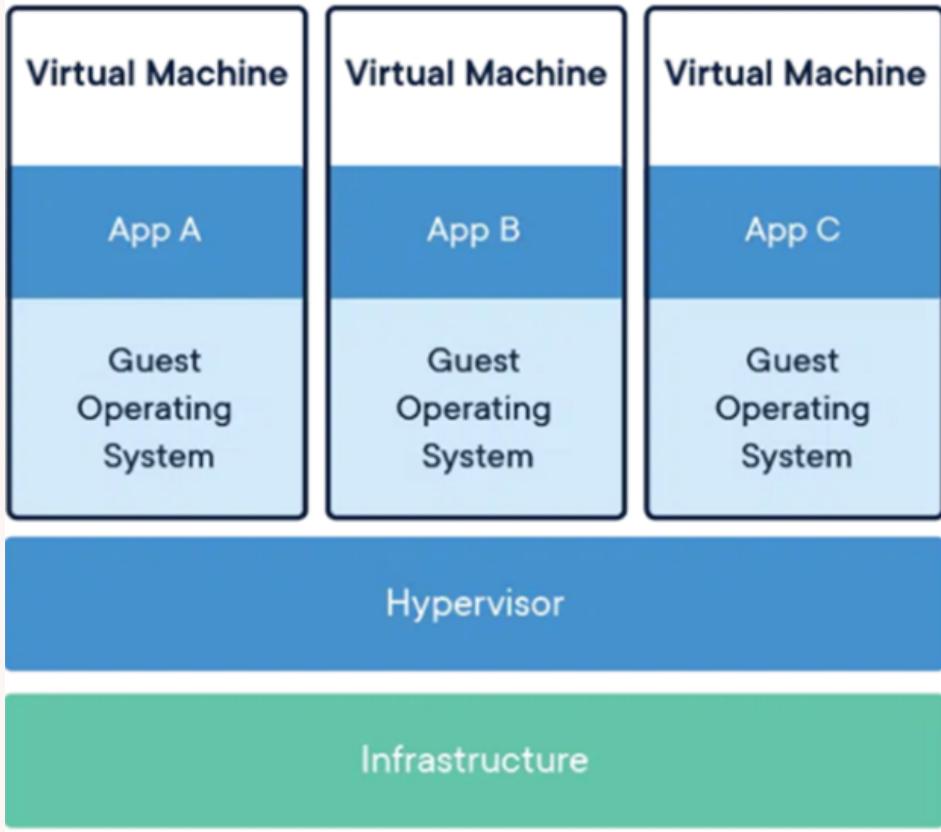
- 하나의 서버에서 여러 애플리케이션(서비스)을 구동하기 위한 환경을 OS 수준의 가상화로 구현

소프트웨어 배포

- 소프트웨어를 가상머신으로 패키징하여 해당 소프트웨어가 최적의 환경에서 동작할 수 있도록 지원

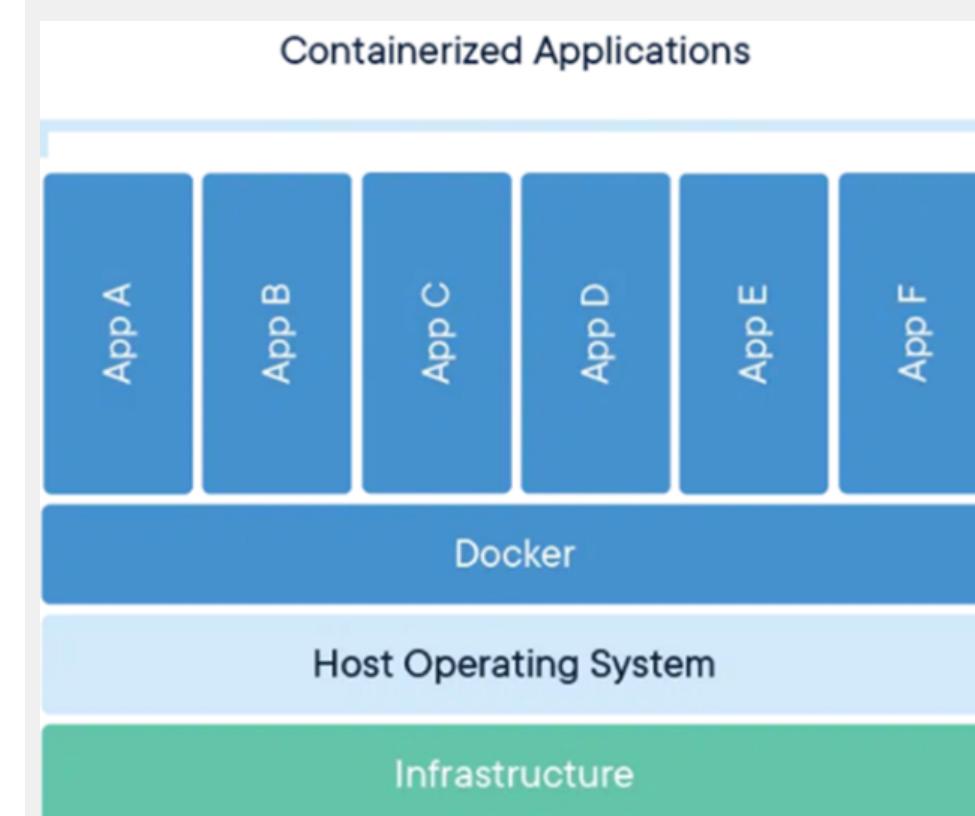
Docker container vs Virtual Machine

VM(가상머신)



- 시스템 자원 가상화 및 독립된 공간 생성 작업은 하이퍼바이저를 거침
→ 성능 손실이 큼
- 가상머신은 게스트 OS를 사용하기 위한 라이브러리, 커널 등을 포함
→ 배포할 때 용량이 큼

Docker container



- 가상화된 공간을 생성할 때 리눅스 자체 기능을 사용하여 프로세스 단위의 격리 환경 생성
→ 성능 손실이 없음
- 가상머신과 달리 커널을 공유해서 사용
- 컨테이너에는 라이브러리 및 실행 파일만 존재
→ 용량이 작음

기존 소프트웨어 배포 문제점

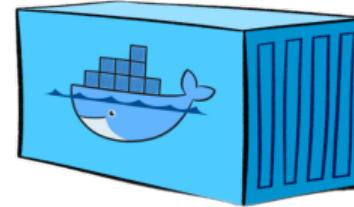
배포조건



기존 소프트웨어 배포조건

- 각 서버에 동일한 환경 요구

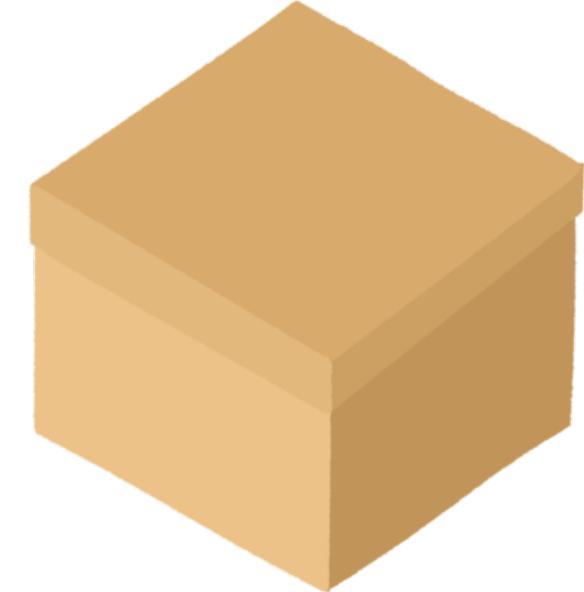
컨테이너 등장



컨테이너의 필요성

- 최적의 소프트웨어 동작을 위한 환경 필요
- 컨테이너 사용 시작

Docker container



정의

- 2013년 Docker 엔진과 함께 출시
- 사용자가 기본 시스템에서 애플리케이션을 분리할 수 있는 가상화된 런타임 환경

특징

- 표준화된 환경
- 간편한 공유
- 자율성

기존 소프트웨어 배포 문제점

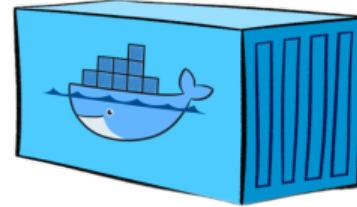
배포조건



기존 소프트웨어 배포조건

- 각 서버에 동일한 환경 요구

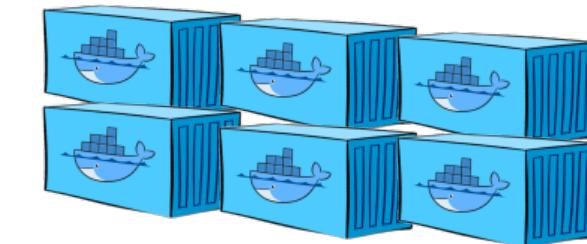
컨테이너 등장



컨테이너의 필요성

- 최적의 소프트웨어 동작을 위한 환경 필요
- 컨테이너 사용 시작

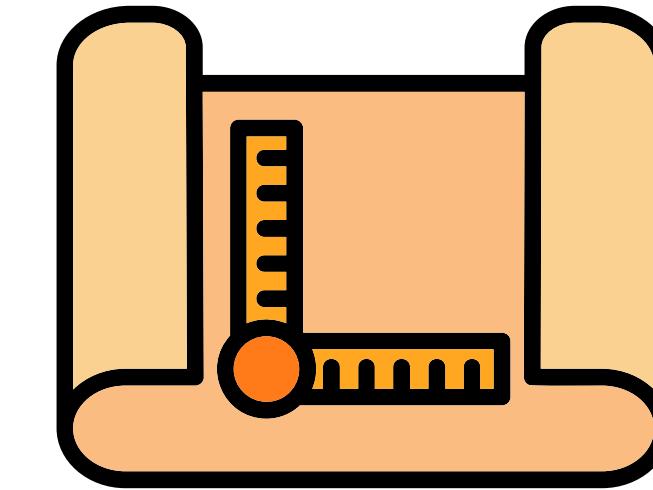
문제점 발생



컨테이너 관리

- 격리환경 구성 과정이 난이도가 높고 복잡함
- 점차 늘어나는 컨테이너 관리에 어려움 발생

Docker image

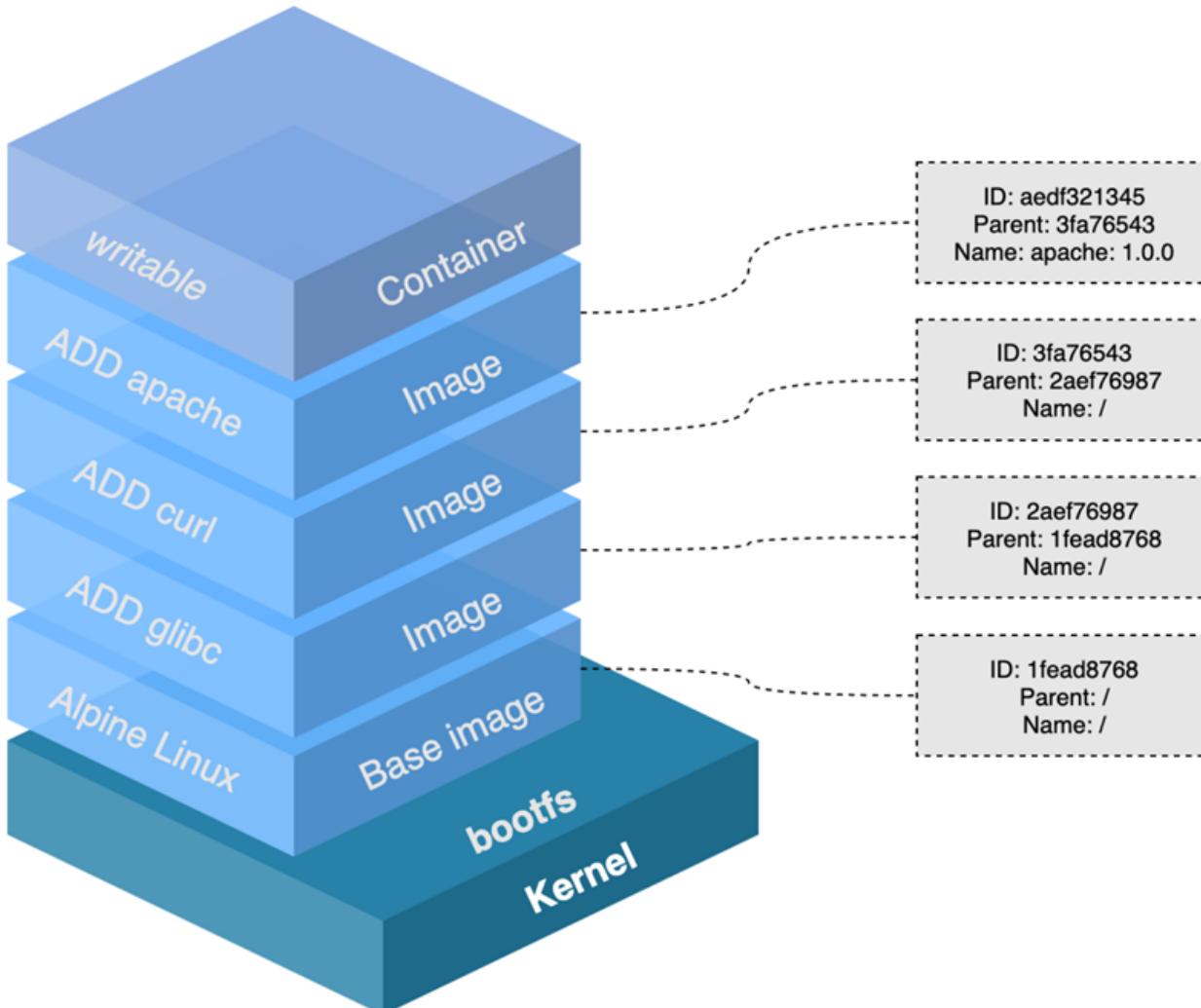


정의

- 런타임 환경을 위한 일종의 템플릿

특징

- 읽기 전용으로 특정 시점의 어플리케이션과 가상 환경을 나타냄
- 소스코드, 라이브러리 등 도구 및 응용 프로그램을 실행하는데 필요한 기타 파일을 포함
- 이러한 일관성은 개발자가 안정적이고 균일한 조건에서 소프트웨어를 테스트하고 실험할 수 있도록 함
- 도커 이미지는 여러 개의 레이어로 구성될 수 있으며 각각은 다르지만 이전 레이어에서 비롯됨



Docker image와 container

- 컨테이너가 존재하려면 이미지를 실행하여야 하는 반면, 이미지는 컨테이너 없이 존재 가능
- 각각의 고유한 데이터와 상태를 가진 동일한 이미지에서 여러 개의 컨테이너를 만들 수 있음
- 컨테이너는 이미지에 종속되어 런타임 환경을 구성하고 어플리케이션을 실행하는 데 사용되어 실행 중인 이미지의 인스턴스로 간주함
- 객체지향 프로그래밍에서의 클래스 <-> 인스턴스 관계와 비슷하다고 생각하면 됨

03

쿠버네티스

등장배경
구성요소

Kubernetes 등장배경

▶ DOCKER의 문제점

컨테이너 수 증가에 따른 관리의 어려움

서버에 대한 종속성

- 하드웨어 문제 발생 시,

해당 서버에 등록된 모든 컨테이너 작동 중단

트래픽 이슈 발생 시 대처방안 필요

해결책: KUBERNETES

자동화된 복구

- 시스템이 영향을 받을 때마다 컨테이너의 상태 감지
- 비정상적인 경우 자체적으로 교체

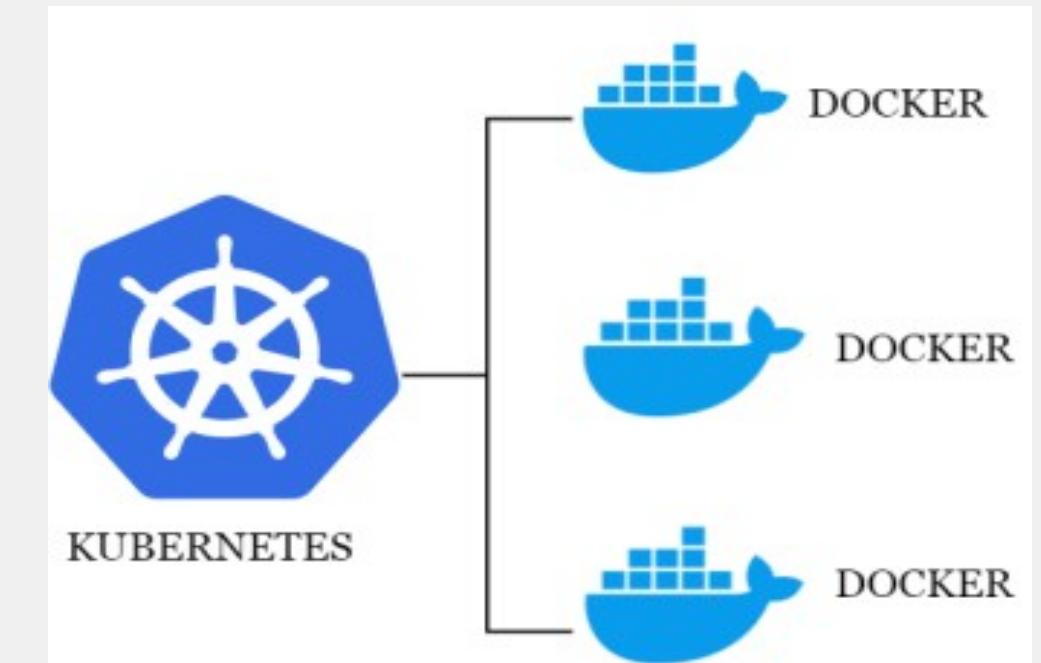
로드 밸런싱

- 문제 발생 시 새로운 컨테이너 생성
- 트래픽 증가 시, 자동으로 네트워크 트래픽 조절
- 트래픽 감소 시, 지정해둔 최소 숫자로 컨테이너 수 조절

Kubernetes

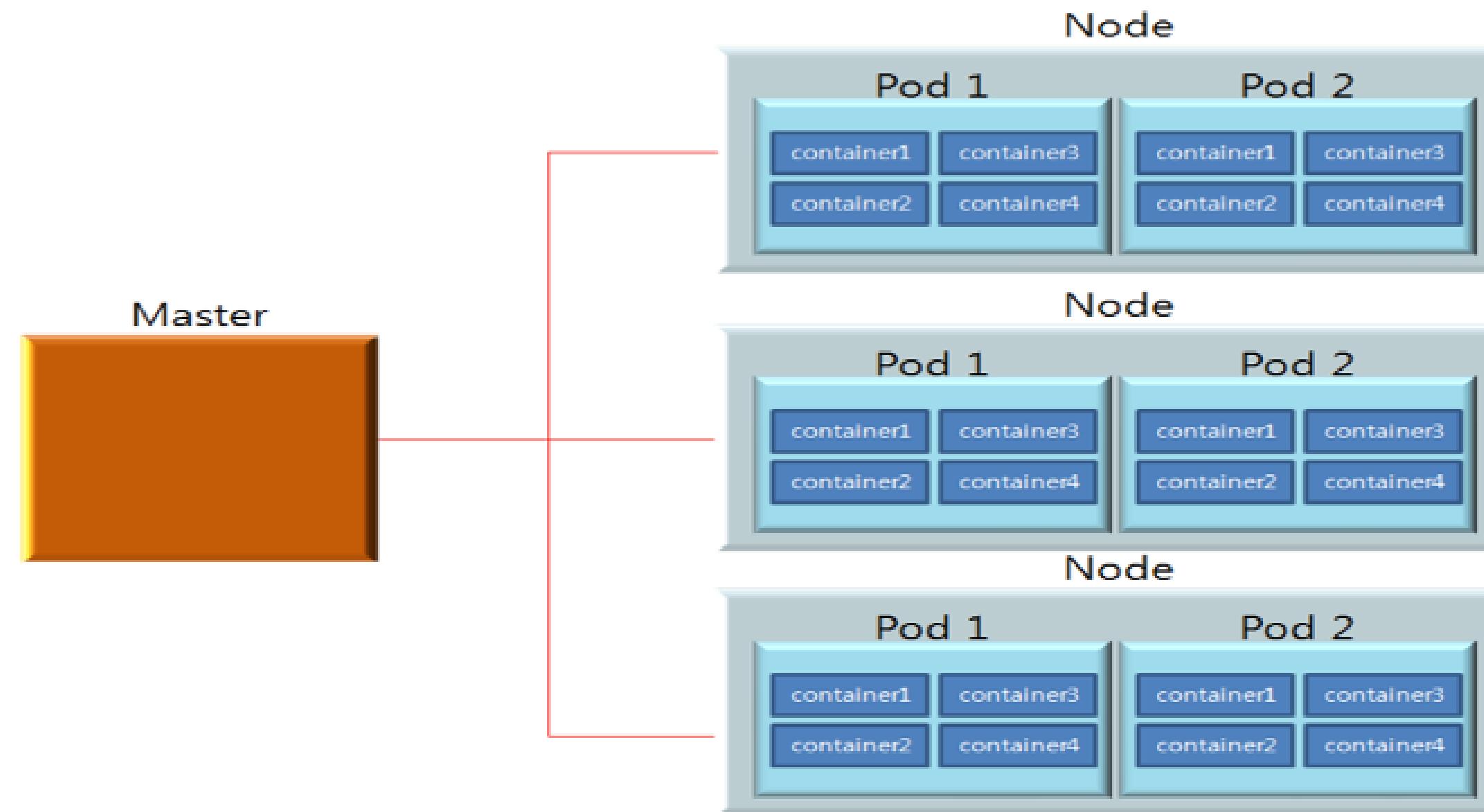
K8S

용어	뜻
컨테이너	앱이 구동되는 환경까지 감싸서 실행할 수 있도록 하는 격리 기술
컨테이너 런타임	컨테이너를 다루는 도구
도커	컨테이너를 다루는 도구 중 가장 유명한 것
쿠버네티스	컨테이너 런타임을 통해 컨테이너를 오케스트레이션 하는 도구
오케스트레이션	여러 서버에 걸친 컨테이너 및 사용하는 환경 설정을 관리하는 행위



- 쿠버네티스는 하나의 컨테이너 오케스트레이션 툴
- MSA는 복잡한 애플리케이션을 독립적인 마이크로서비스로 분할하여 각각을 개발, 배포, 확장할 수 있도록 하는 소프트웨어 아키텍처 패턴임
- 각 서비스는 경량화된 통신으로 상호 작용하며, 각각이 개별 데이터 스토어를 가질 수 있음
- 쿠버네티스는 이러한 아키텍처 패턴을 지원하여 MSA를 구현하는데 있어, 각 서비스를 쉽게 관리하고 확장할 수 있도록 도와줌

Kubernetes 클러스터



이렇게 구성된 쿠버네티스 클러스터는 분산 환경에서 컨테이너화된 애플리케이션을 효율적으로 관리하고 배포할 수 있도록 해줌.

클러스터 구성요소 - 마스터

① 마스터

- 쿠버네티스 클러스터의 중앙 관리자 역할
- 클러스터의 모든 일을 총괄, 조정

주요 기능

- **노드 제어, 감독** - 각 워커 노드의 작업 상태를 지속적으로 모니터링하여 필요한 작업을 할당
- **클러스터 환경 설정 저장** - 클러스터의 상태를 유지하고 리소스를 효율적으로 활용
- **클러스터 전체를 관리**

클러스터 구성요소 - 노드

② 노드

- 클러스터 내에서 실제로 애플리케이션을 실행하는 서버

주요 기능

- **컨테이너 실행, 관리** - 컨테이너를 실행하기 위한 환경 제공, 컨테이너를 시작하고 중지
- **워커 노드** - 마스터 노드로부터 할당받은 작업을 처리하고 클러스터의 일부로서 동작
- **파드 호스팅** - 파드가 필요로 하는 리소스 제공, 파드 내부에서 컨테이너를 실행할 수 있는 환경 제공

클러스터 구성요소 - 파드

③ 파드

- 컨테이너화된 애플리케이션을 실행하는 가장 작은 단위

주요 기능

- **컨테이너 그룹** - 하나 이상의 컨테이너를 포함
- **네트워크 공유** - 파드 내부의 컨테이너들은 통신 가능
- **저장소 공유** - 컨테이너 간에 데이터 공유, 중요한 정보 저장 가능

— 04 —

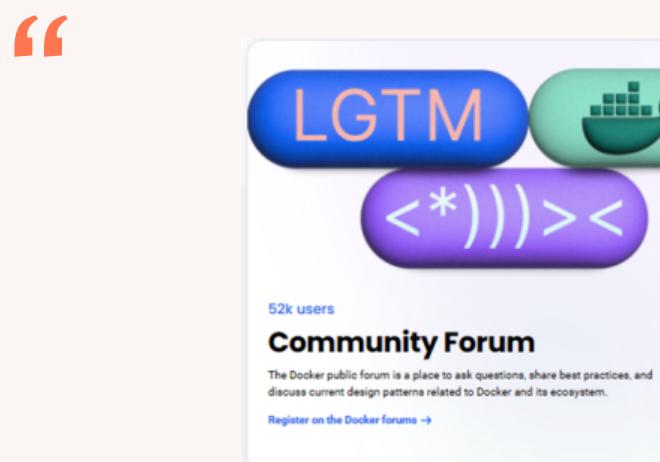
커뮤니티

도커 커뮤니티

쿠버네티스 커뮤니티

Official communities

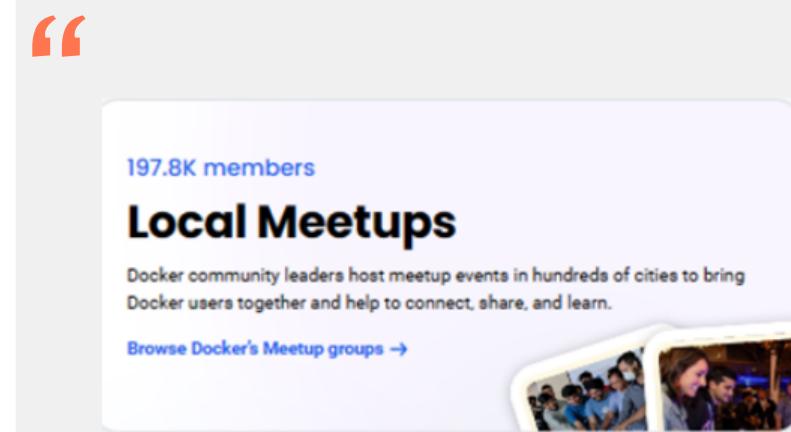
Docker 커뮤니티 ①



Community Forum

- 문제 발생 시 질문
- 포럼을 통한 경험자들의 답변(경험 및 팁)

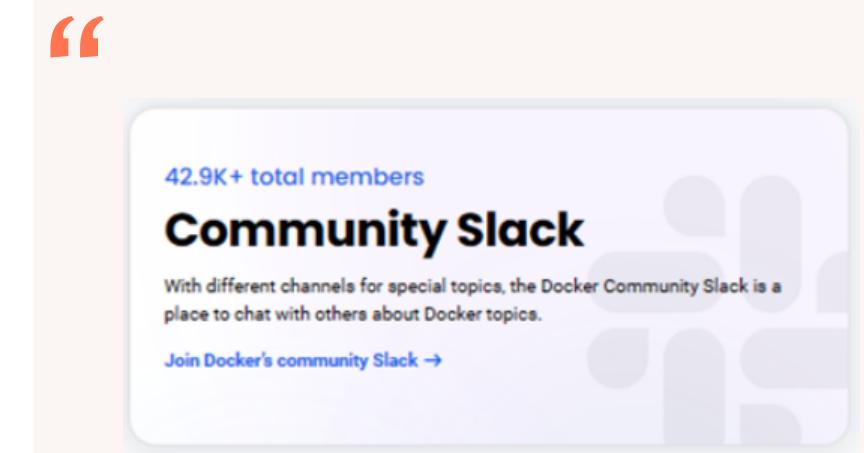
Docker 커뮤니티 ②



Local Meetup

- 개발자들 간 커뮤니티
- 사용자 그룹 간 Meetup 이벤트

Docker 커뮤니티 ③



Community Slack

- Docker Slack 채널 탐색
- 사례) #buildkit 채널
- 메신저 기능

Community Forum

FORUM

Welcome to the Docker Community Forums!

This is a public forum for users to discuss questions and explore current design patterns and best practices about Docker and related projects in the Docker Ecosystem.

To participate, just log in with your Docker Hub account. Make sure to also review our [Community Guidelines](#), [Terms of Service](#), and [Privacy Policy](#).

When posting issues or feedback, make sure to remove any sensitive information and please provide the following:

- Issue type
- OS Version/build
- App version
- Steps to reproduce

Category	Topics	Latest
Announcements	52	 My first steps - docker under Ubuntu, build an image from given Dockerfile General docker
General Discussions	13.5k	 CIFS volume for a Windows domain user Docker Desktop for Windows cifs
Open Source Projects	5.0k	 After running the container, my connection with the server is disconnected General Discussions

- 문제 발생 시, 해당 문제 경험자들에게 질문
- 경험자들의 해결책 및 팁 공유
- 질문 이어나가기 가능
- 커뮤니티 포럼에서 자유롭게 의견 피력 및 답변 가능

질문하기

After running the container, my connection with the server is disconnected

General Discussions



hack3rcon

Apr 24

1 / 3
Apr 24

Hello,
The docker-compose.yml file is as follows:

```
services:  
  banner:  
    container_name: Project  
    build:  
      context: /data/project  
      dockerfile: Dockerfile  
    ports:  
      - "3000:3000"  
    environment:  
      - PORT=3000  
    command: npm start
```

- 현재 자신이 겪고 있는 문제상황을 포럼에 공유
- 즉시 커뮤니티에 속한 멤버에게 노출

답변하기

bluepuma77



It seems your LAN connection (ens192) suddenly has a private IP (172.20.2.100).

Not sure how that can happen, it shouldn't be directly related to that container.

Did you mess with Docker config, network or firewall settings on host?



hack3rcon

Hello,
Thank you so much for your reply.
The 172.20.2.100 is the IP address of the host NIC.

- 해당 문제를 경험했던 커뮤니티 사용자가 질문자에게 즉각적으로 회신 가능

Local Meetup

DOCKER의 MEETUPS 이벤트

Upcoming events

In person events Online events

	WED, MAY 1 · 7:00 AM GMT+9 Docker e Inteligência Artificial Docker Rio de Janeiro • Rio de Janeiro, BR	21 attendees
	WED, MAY 8 · 1:00 AM GMT+9 Introduction to Docker + KubeCon Updates Docker Freiburg and Black Forest • Freiburg, DE	4 attendees
	SAT, MAY 18 · 12:30 PM GMT+9 Kubetools Day 2.0 "Show-n-Tell" Docker Bangalore • Bangalore, IN	120 attendees

- Community Forum과 달리 Docker를 사용하는 개발자들끼리 직접 모여 의견 공유

전 세계 DOCKER 사용자 그룹 현황

A world map with red dots representing Docker user groups across 61 countries.

- Docker그룹은 한 국가 내 여러 그룹 존재 가능
- 현재 61개국에 167개 그룹 존재

Groups

	Docker Shanghai SHANGHAI, CN Learn, Collaborate & Dockerize! Meet other developers and ops engineers in your community that are using and learning about Docker. Docker is an open platform that helps you build, ship and run applications anytime and anywhere. Developers use Docker to modify cod... 895 members · Public
	Docker Tokyo TOKYO, JP 2017年11月以降のイベントについては、このページではなく https://dockerjp.connpass.com/ にて参加を募集しています。 (We have been using https://dockerjp.connpass.com/ for managing RSVPs since November 2017)...
	Docker Taipei TAIPEI, TW Learn, Collaborate & Dockerize! Meet other developers and ops engineers in your community that are using and learning about Docker. Docker is an open platform that helps you build, ship and run applications anytime and anywhere. Developers use Docker to modify cod... 2383 members · Public

- 주변 인접국(중국, 일본, 대만) 내 Docker그룹 존재
- 현재 우리나라에는 공식적으로 존재하는 Docker그룹은 없음

Community Slack

DOCKER SLACK 채널 탐색

- 참여할 채널 탐색 가능
- 관심있고 궁금한 채널에 직접 참여하여 의견 교류 가능

#BUILDKIT 채널

buildkit concurrent, cache-efficient, and Dockerfile-agnostic builder toolkit - <https://github.com/moby/buildkit>

4월 24일 수요일

Felix de Souza 오전 2:23
finally getting around to rolling out the `rewrite-timestamp` internally together with the containerd change (simplifies a lot of workarounds for us), and curious about the constraint: failed to solve: exporter option "rewrite-timestamp" conflicts with "unpack" the thing to actually get out of it is to set `store=false`. Is there any reason why we can't load the rewritten timestamps into docker's containerd image store?

Galileo 오전 3:44
Wondering if someone could explain what the point of the `attrs` and `caps` actually are on Ops in general. Maybe to scope this question better:
What does the `local.followpaths` attribute do? It gets implicitly created in
`RUN --mount=type=bind,target=/context,source=/upper_level/lower_level`
and gets set as an array with the contents being `local.followpaths=[\"upper_level/lower_level\"]`. I was actually confused until a few minutes ago and thought that the followpaths were responsible for mounting at a specific point in the context (it's actually the `selector` on the mount itself, rather than anything to do with the source op) (번역됨)

Galileo 오전 6:37
Is there a difference to an ExecOp vs. an equivalent FileOp?

B I S | ⌂ | ⌂ ⌂ | ⌂ ⌂ | ⌂ ⌂ | ⌂ ⌂

#buildkit에 메시지 보내기

- 관련 피드백과 토론 내용이 저장됨
- 틀린 피드백일 시, 올바른 피드백으로 수정하는 커뮤니티의 모습

메신저 기능

- ▼ 채널
 - # announcements
 - # buildkit
 - # general
- 🔍 모든 채널 찾아보기
- ▼ 다이렉트 메시지
-  Lim Yu Han 나
- 🔍 직장 동료 추가

- 직장 동료를 추가하여 동료가 초대를 받으면, 그 동료에게 메시지를 보낼 수 있는 SNS 기능 존재
 - 그 사람에게 직접 메시지를 보내는 것이기 때문에 다른 사람은 볼 수 없는 보안성 보장

Blog

K8S의 BLOG

Posts in 2024

RSS 

[Spotlight on SIG Architecture: Code Organization](#)

Thursday, April 11, 2024 in Blog

This is the third interview of a SIG Architecture Spotlight series that will cover the different subprojects. We will cover SIG Architecture: Code Organization. In this SIG Architecture spotlight I talked with Madhav Jivrajani (VMware), a member of ...

[Read more](#)

[Using Go workspaces in Kubernetes](#)

Tuesday, March 19, 2024 in Blog

The Go programming language has played a huge role in the success of Kubernetes. As Kubernetes has grown, matured, and pushed the bounds of what “regular” projects do, the Go project team has also grown and evolved the language and tools. ...

[Read more](#)

- k8s 자체 내에서 운영하는 블로그
- 정보의 정확성이 뛰어남
- 궁금한 내용이 있을 때 검색 가능

Mailing List

SIG로 구성된 K8S 커뮤니티

Special Interest Groups ↗

Name	Label	Chairs	Contact	Meetings
API Machinery	api-machinery	* David Eads, Red Hat * Federico Bongiovanni, Google	* Slack * Mailing List	* Kubebuilder Meeting: Thursdays at 11:00 PT (Pacific Time) (biweekly) * Regular SIG Meeting: Wednesdays at 11:00 PT (Pacific Time) (biweekly)
Apps	apps	* Janet Kuo, Google * Kenneth Owens, Snowflake * Maciej Szulik, Red Hat	* Slack * Mailing List	* Regular SIG Meeting: Mondays at 9:00 PT (Pacific Time) (biweekly)

- k8s의 커뮤니티 활동은 SIG (Special Interest Group)으로 구성됨
- 이는 k8s 커뮤니티의 정보 정확성을 높여줌

K8S의 MAILING LIST

K8s API Machinery SIG

소유자 및 관리자에게 문의

4457 중 1~30개 < >

Explicit non-goals:

1. Particular APIs or API objects will not be discussed here, unless as a concrete example of something that is in-scope.
2. Other forms of extensibility are worthy goals but not within the scope of this SIG.

For more explanation go [here](#) (doc shared with this group, kubernetes-dev@, and kubernetes-announce@)

...	Todd Treece	Running Extension API Servers Outside of Kubernetes — Hi, It was suggested that I reach out to this group a...	3월 12일	☆
...	Haibing Zh..., David Protasow...	Two Lane Queue in client-go/workqueue — If you haven't seen Knative has implemented a two-lane queue in ...	3월 4일	☆
...	David Eads	Exception for KEP-4346: client-go: Add informer metrics — Enhancement name: KEP-4346: client-go: Add inf...	2월 10일	☆
...	Rob Scott, ... Jonathan Innis	AdditionalPrinterColumns Array Support — We just came up on this in Karpenter and I was still thinking that ...	2월 5일	☆
...	Joe Betz	KEP-4358: Custom Resource Field Selectors — All, I've opened https://github.com/kubernetes/enhancement...	1월 12일	☆
...	Federico Bongio..., ... Leila Ja...	SIG API Machinery 2023 Contributor Awards, congratulations! — Congratulations Mike, @MikeSpreitzer, and ...	23. 12. 5.	☆
D	David Oppen..., Federico Bon...	limit on total number of concurrency shares for API Priority & Fairness? — Hi David, thanks for your question...	23. 11. 29.	☆

- 수신자와 송신자만 확인 할 수 있는 일반적 메일과 다름
- k8s 커뮤니티는 그룹 이메일을 사용하여 참여한 사람들 모두 내용을 볼 수 있고 답변도 할 수 있도록 커뮤니티를 구축함

Community Forum 및 SNS

K8S의 COMMUNITY FORUM

Category	Topics	Latest
Announcements	344	 Please refrain from posting CKA/CKAD exam content or cheatsheets here ■ General Discussions ■ cncf Dec 2023
Announcements from around the Kubernetes project related to core Kubernetes.		
■ Ecosystem		
General Discussions	5.6k	 HackerOne Bug Bounty Program ■ General Discussions May 2020
Got something to say about Kubernetes? Hop in here. Links to interesting cloud-native blog posts encouraged.		
■ Windows ■ microk8s		
Regional Discussions	116	 Welcome to Discuss Kubernetes Apr 2022
The Regional Discussions Category is intended for those users that belong to a specific region or share a common language to openly interact and connect with each other in their native language. If you are interested in establishing a new regional board, please reach out to one of the admins .		
■ Chinese ■ Ukrainian ■ Italian ■ German ■ Spanish ■ Arabic		
Events and Meetups	86	 Issue in installing kubernetes on linux 20.04 ■ General Discussions 5m
Manage multiple kubectl port-forward in the menu bar	0	 General Discussions development network 8h

- Docker와 마찬가지로 왼쪽의 카테고리에 자주 하는 질문을 모아놓아 사용자의 접근성을 높임

K8S의 SNS 활동

토론

우리는 대화를 많이 합니다. 아래의 플랫폼에서 우리를 찾아 대화에 참여하세요.


커뮤니티 포럼▶
 문서, 트러블슈팅 및 그 외 더 많은 것을 소개하는 주제 기반 기술 토론이 열리는 곳입니다.


트위터▶
 #kubernetesio
 블로그 게시물, 이벤트, 뉴스, 아이디어에 대한 실시간 소식 들입니다.


GITHUB▶
 모든 프로젝트와 이슈 추적, 그리고 물론 코드도 있습니다.


SERVERT FAULT▶
 Server Fault에 쿠버네티스 관련 토론들이 있습니다. 질문을 하거나, 질문에 답해 보세요.


SLACK▶
 170개 이상의 채널이 있으며, 필요에 맞는 채널을 찾을 수 있습니다.
 ▶ 초대가 필요하신가요?

- K8S의 SNS 활동

—appendix—

사용방법

Docker 설치 및 사용방법
Kubernetes 기본 명령어

이미지 관리

*사용방법

```
C:\#Userst#User>docker image ls
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ex22_original2  latest   a3f3a8eee801  41 hours ago  147MB
httpd           latest   67c2fc9e3d84   3 weeks ago   147MB
redmine         latest   e7d6aa421df2  3 weeks ago   632MB
mysql           latest   6f343283ab56   4 weeks ago   632MB

C:\#Userst#User>docker image rm httpd
Untagged: httpd:latest
Untagged: httpd@sha256:36c8c79f900108f0f09fd4148ad35ade57cba0dc19d13f3d15be24ce94e6a639
Deleted: sha256:67c2fc9e3d849b21a35b4c96f5ad8ec4dc9b73ca44c56537039e8c6f3054db0a

C:\#Userst#User>docker image ls
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ex22_original2  latest   a3f3a8eee801  41 hours ago  147MB
redmine         latest   e7d6aa421df2  3 weeks ago   632MB
mysql           latest   6f343283ab56   4 weeks ago   632MB

C:\#Userst#User>docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ex22_original2  latest   a3f3a8eee801  41 hours ago  147MB
redmine         latest   e7d6aa421df2  3 weeks ago   632MB
mysql           latest   6f343283ab56   4 weeks ago   632MB
```

- 이미지 검색

docker search [이미지명]

도커 허브에서 원하는 이미지를 검색

예시) docker search httpd

- 이미지 다운로드

docker pull [이미지명]

도커 허브에서 이미지를 다운로드

예시) docker pull httpd

- 이미지 목록 확인

docker images

현재 시스템에 다운로드된 이미지 목록을 확인

- 이미지 삭제

docker rmi [이미지명]

더 이상 필요 없는 이미지를 삭제

*사용방법

```
명령 프롬프트
5f

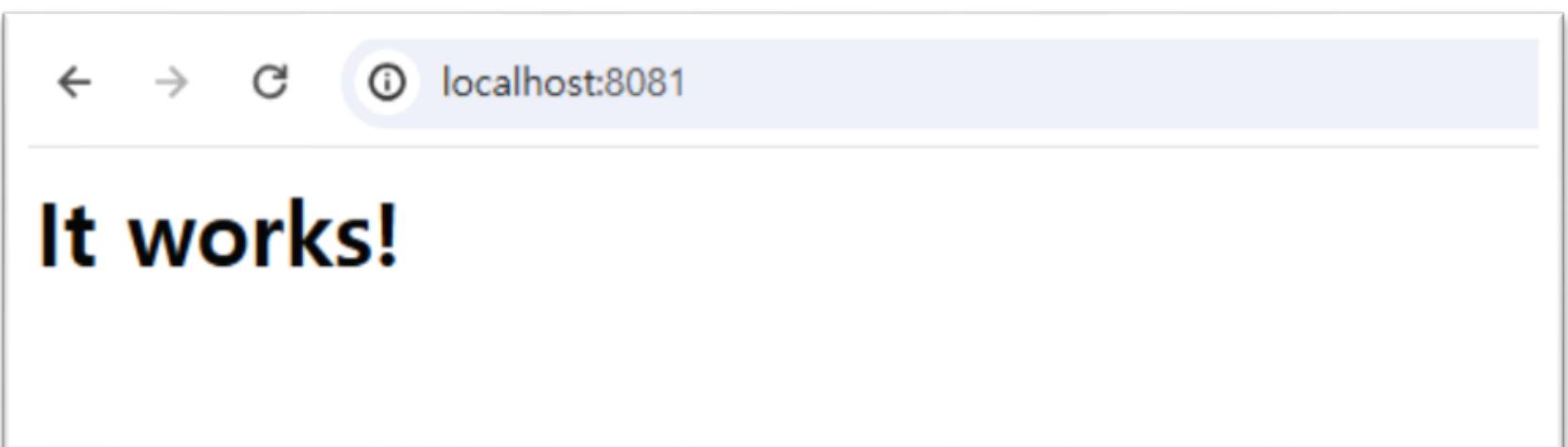
C:\Users\User>docker run --name apa000ex1 -d httpd
10ff6bc535e5448935f485b179661e0dc2ba2c2f7a5a316c91c5144be3b2e154

C:\Users\User>docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
10ff6bc535e5 httpd "httpd-foreground" 6 seconds ago Up 5 seconds 80/tcp apa000ex1

C:\Users\User>docker run --name apa000ex2 -d -p 8081:80 httpd
a53acd0b5e5096e4ced8385e6b4ed759d20a78874b79d3e04e6c331fb39f6e5a7

C:\Users\User>docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a53acd0b5e509 httpd "httpd-foreground" 3 seconds ago Up 2 seconds 0.0.0.0:8081->80/tcp apa000ex2
10ff6bc535e5 httpd "httpd-foreground" 4 minutes ago Up 4 minutes 80/tcp apa000ex1
```

Docker 컨테이너 확인 및 관리 예시



컨테이너 관리-1

● 컨테이너 실행

docker run [옵션] [이미지명]

이미지를 실행하여 컨테이너를 생성

● 목록 확인

docker ps

현재 실행 중인 컨테이너 목록을 확인

● 모든 목록 확인

docker ps -a

중지된 컨테이너까지 모두 확인

● 컨테이너 중지

docker stop [컨테이너 이름]

*사용방법

명령 프롬프트

```
5f  
C:\Users\user>docker run --name apa000ex1 -d httpd  
10ff6bc535e5448935f485b179661e0dc2ba2c2f7a5a316c91c5144be3b2e154  
  
C:\Users\user>docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
10ff6bc535e5 httpd "httpd-foreground" 6 seconds ago Up 5 seconds 80/tcp apa000ex1  
  
C:\Users\user>docker run --name apa000ex2 -d -p 8081:80 httpd  
a53ac0b5e5096e4ced8385e6b4ed759d20a78874b79d3e04e6c331fb39f6e5a7  
  
C:\Users\user>docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
a53ac0b5e509 httpd "httpd-foreground" 3 seconds ago Up 2 seconds 0.0.0.0:8081->80/tcp apa000ex2  
10ff6bc535e5 httpd "httpd-foreground" 4 minutes ago Up 4 minutes 80/tcp apa000ex1
```

Docker 컨테이너 확인 및 관리 예시



컨테이너 관리-2

- 컨테이너 시작

docker start [컨테이너 이름]

- 컨테이너 로그 확인

docker logs [컨테이너 이름]

- 컨테이너 삭제

docker rm [컨테이너 이름]

- 도커 컨테이너 이름 지정

docker run --name [컨테이너 명] [이미지명]

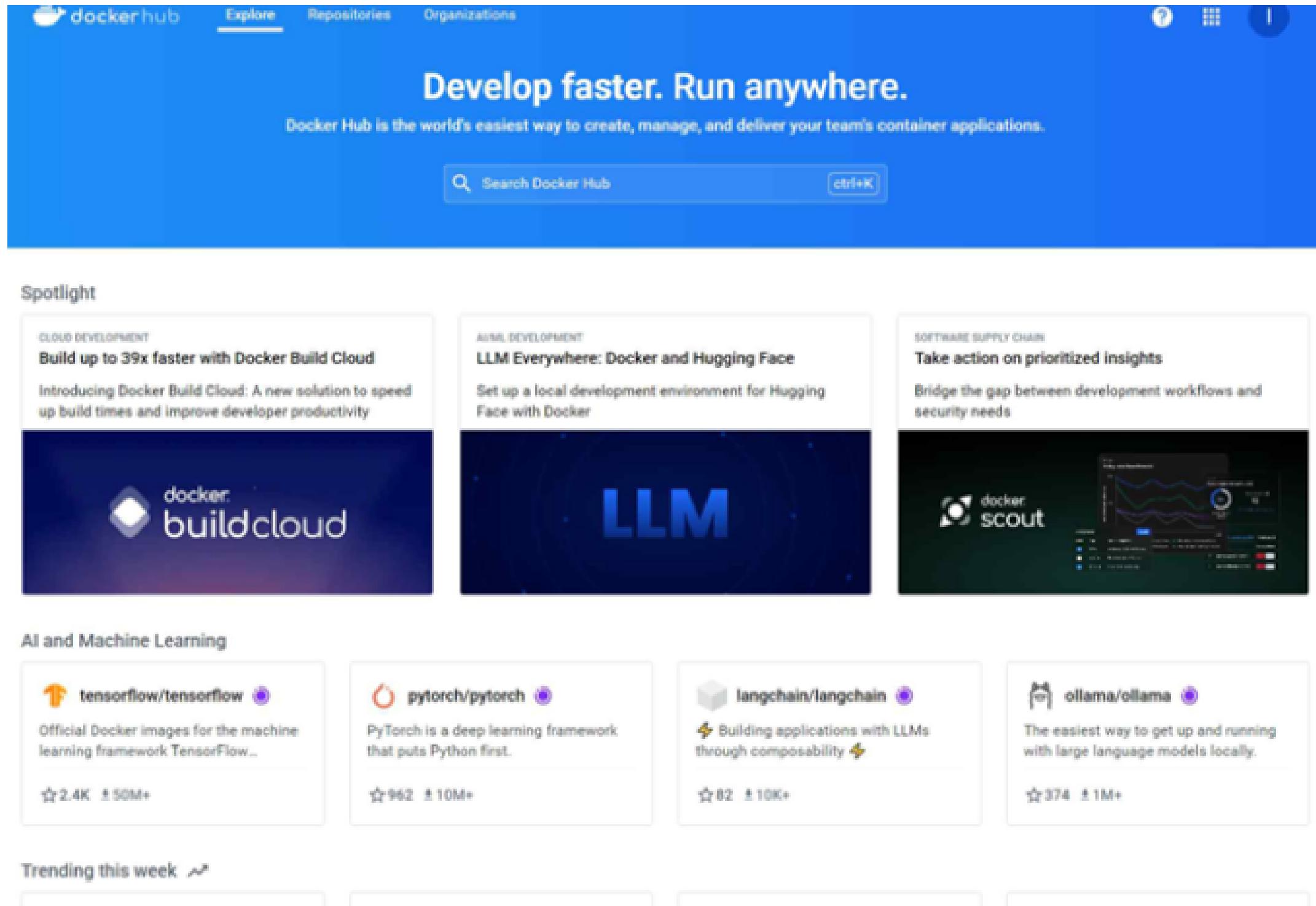
예시: docker run --name ws2 httpd

- 컨테이너에 다시 연결

docker attach [컨테이너 이름]

예시: docker attach my-ubuntu

* 사용방법



이미지 사용방법 ①- Docker Hub에서 이미지를 다운로드 받아 사용

● 도커 허브 접속

- 도커 허브 접속 후 explore를 클릭
- 메인 화면에서 필요한 이미지를 찾거나
- 'Search Docker Hub'로 필요한 이미지를 검색
ex) httpd 검색

*사용방법

The screenshot shows the Docker Hub interface for the 'httpd' image. At the top, there's a navigation bar with 'docker hub', 'Explore', 'Repositories', and 'Organizations'. Below it, a search bar contains 'httpd'. The main content area shows the 'httpd' image card, which includes a logo of a feather, the name 'httpd', a 'Docker Official Image' badge, a rating of 10+, and 4.7K stars. It also mentions 'The Apache HTTP Server Project'. A 'Recent Tags' sidebar lists tags like 'latest', 'bookworm', 'alpine3.19', 'alpine', '2.4.59-bookworm', '2.4.59-alpine3.19', '2.4.59-alpine', '2.4', '2', 'latest', '2.4.59-bookworm', '2.4-bookworm', '2-bookworm', and 'bookworm'. Below the card, sections include 'Quick reference', 'Supported tags and respective Dockerfile links', and 'Quick reference (cont.)'. The 'Quick reference' section lists 'Maintained by: the Docker Community' and 'Where to get help: the Docker Community Slack, Server Fault, Unix & Linux, or Stack Overflow'. The 'Supported tags and respective Dockerfile links' section lists tags such as '2.4.59', '2.4', '2', 'latest', '2.4.59-bookworm', '2.4-bookworm', '2-bookworm', and 'bookworm'. The 'Quick reference (cont.)' section lists 'Where to file issues: https://github.com/docker-library/httpd/issues', 'Supported architectures: (more info) amd64, arm32v5, arm32v6, arm32v7, arm64v8, i386, mips64le, ppc64le, s390x', and 'Published image artifact details: repo-info repofs repos/httpd/ directory (history)'.

이미지 사용방법 ②

- Docker Official images
- 도커에서 공식적으로 인증

*사용방법

```
명령 프롬프트
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\YOUNG>docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
13808c22b207: Pull complete
6e9a8835eae4: Pull complete
4f4fb700ef54: Pull complete
b927d001db70: Pull complete
559cc51378ed: Pull complete
d2b091e65160: Pull complete
Digest: sha256:b19cace6539a05579c55fda6be0a873c1d2c2e7392e7c08805141f79852ab07b
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview httpd

C:\Users\YOUNG>docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
httpd           latest        fa0099f1c09d   2 weeks ago   148MB

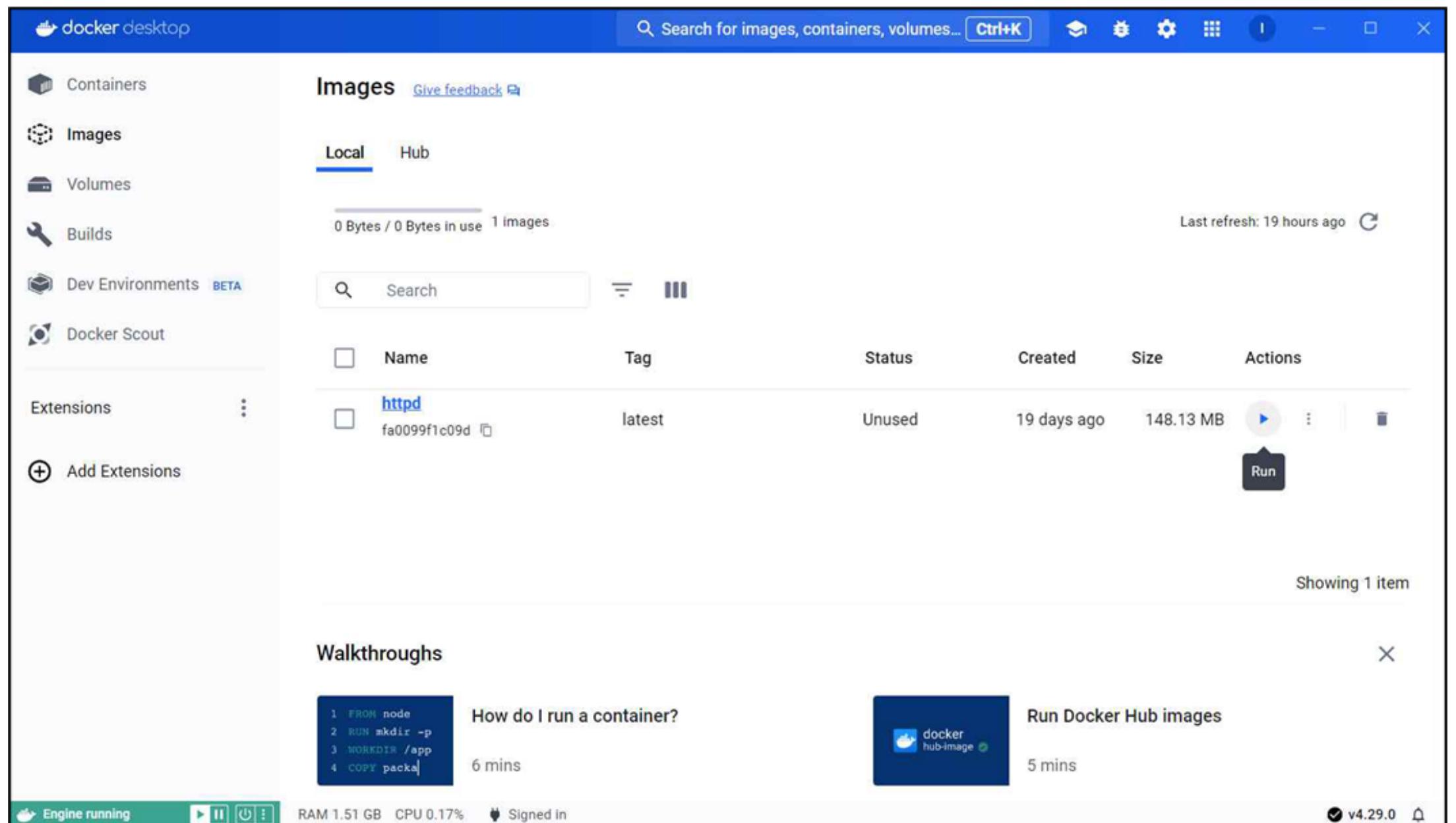
C:\Users\YOUNG>
```

이미지 사용방법 ③

- 명령 프롬프트

- 명령 프롬프트에 ‘docker pull httpd’를 입력

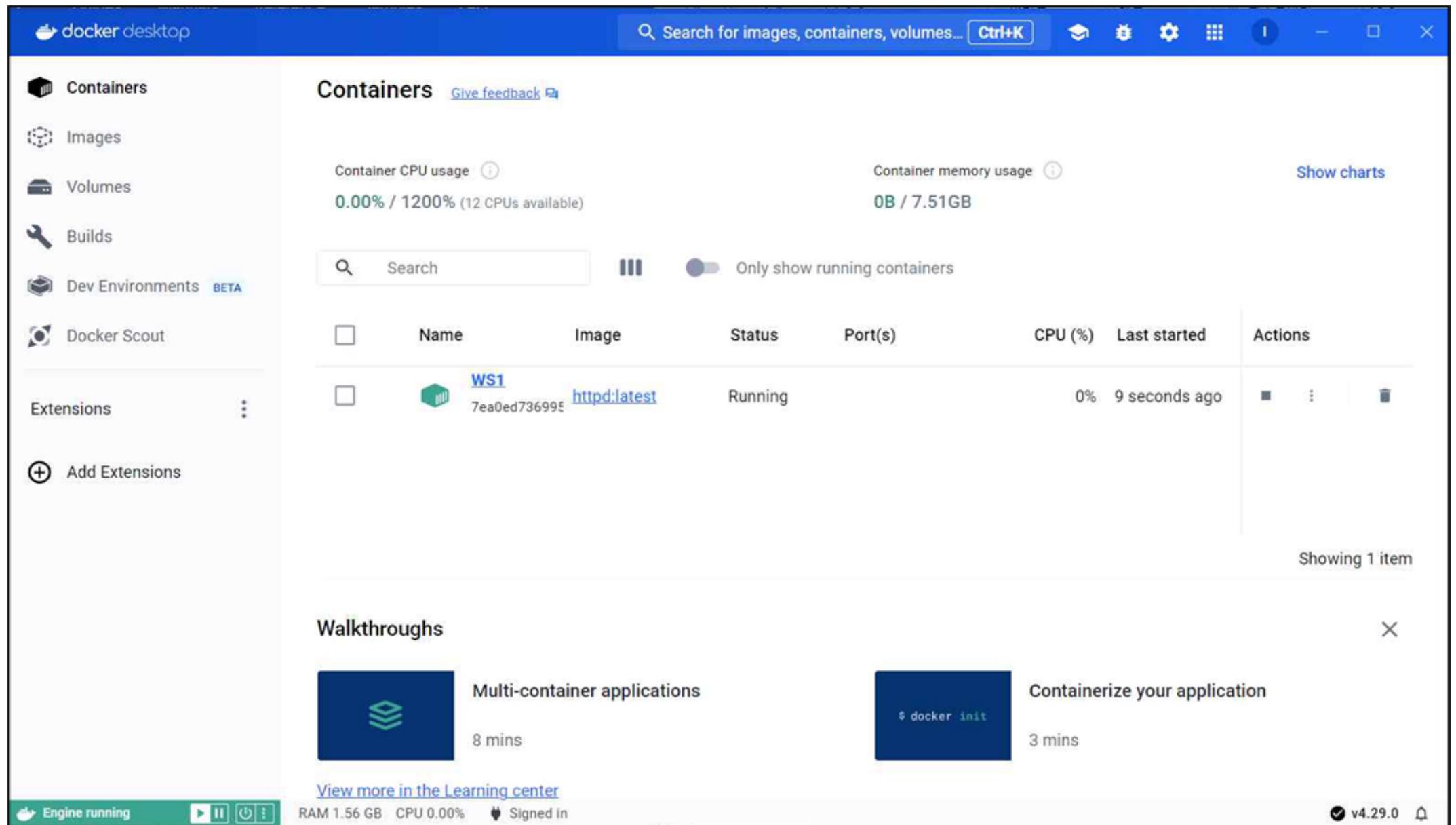
*사용방법



이미지 사용방법 ④

- 이름 및 포트 변경
 - docker desktop에서 images에서 run 버튼 클릭
Optional settings 클릭 시 이름과 포트 등 변경 가능

*사용방법



이미지 사용방법 ⑤

- log 확인
 - 실행하면 images에서 containers로 이동된 것을 확인할 수 있음
 - 컨테이너의 이름 클릭 시 log 등 확인 가능

* 사용방법

The screenshot shows the Docker Hub user profile for 'jyzzang02@kau.kr'. The top navigation bar includes links for 'Explore', 'Repositories', and 'Organizations'. A search bar with the placeholder 'Search Docker Hub' and a 'ctrl+K' keyboard shortcut is also present. The user's profile picture is a blue fingerprint icon. Below the profile picture, the username 'jyzzang02@kau.kr' is displayed with a link to 'Edit profile'. The user is identified as a 'Community User' who joined on April 23, 2024. The main content area shows a single repository: 'icaros85/egoing' by 'icaros85', updated a few seconds ago. The repository has 0 stars and 0 reviews. The repository card includes a preview image and a link to 'Image'. Navigation tabs at the bottom include 'Repositories' (which is selected), 'Starred', and 'Contributed'. A message at the bottom states 'Displaying 1 to 1 repositories'.

이미지를 Docker Hub에 올리는 방법 ①

- 이미지 태그 지정

이미지 태그 형식

[Docker Hub 사용자 이름] / [이미지 이름]:[태그]

예시) Docker Hub 사용자 이름이 'username',

이미지 이름이 'myimage', 태그가 'v1'인 경우

이미지 태그는 [username/myimage:v1]

- 명령어: docker tag [기존 이미지 이름]:[기존 태그] [새 이미지 태그]

예시) docker tag myimage:latest

username/myimage:v1

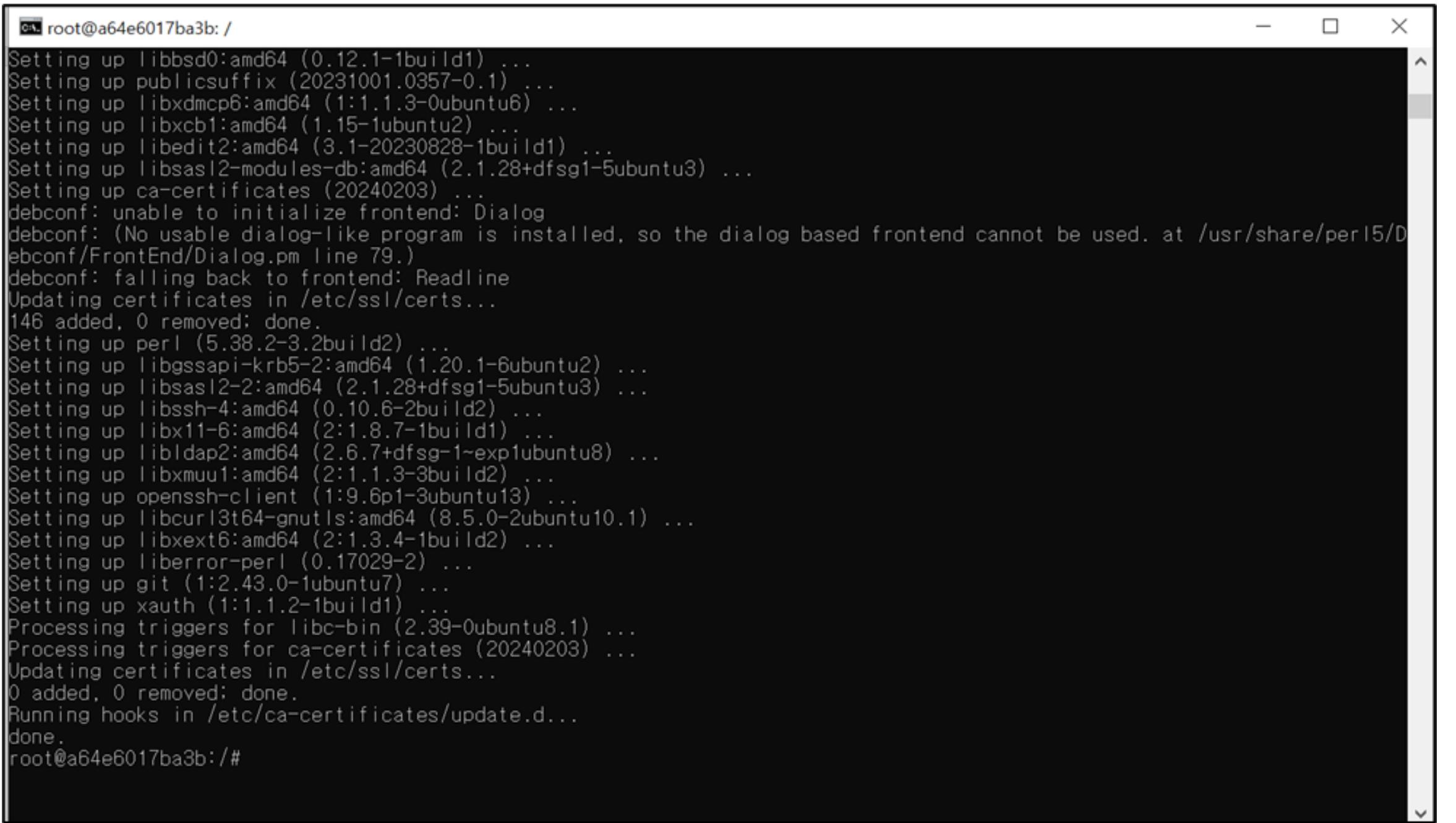
*사용방법

The screenshot shows the Docker Hub profile page for the user 'jyzzang02@kau.kr'. The top navigation bar includes links for 'Explore', 'Repositories', and 'Organizations'. A search bar with the placeholder 'Search Docker Hub' and a keyboard shortcut 'ctrl+K' is present. The user's profile information is displayed, including their email 'jyzzang02@kau.kr', a link to 'Edit profile', and their status as a 'Community User' joined on April 23, 2024. Below the profile, there are tabs for 'Repositories', 'Starred', and 'Contributed', with 'Repositories' being the active tab. It displays '1 to 1 repositories'. A single repository card for 'icaros85/egoing' is shown, created by 'icaros85' a few seconds ago. The repository has 0 stars and 0 forks. A small 'Image' button is at the bottom of the card.

이미지를 Docker Hub에 올리는 방법 ②

- 이미지 푸시
 - 명령어: docker push [이미지 태그]
 - 예시) docker push username/myimage:v1

* 사용방법



The screenshot shows a terminal window with the following text output:

```
root@a64e6017ba3b: /  
Setting up libbsd0:amd64 (0.12.1-1build1) ...  
Setting up publicsuffix (20231001.0357-0.1) ...  
Setting up libxdmcp6:amd64 (1:1.1.3-0ubuntu6) ...  
Setting up libxcb1:amd64 (1.15-1ubuntu2) ...  
Setting up libedit2:amd64 (3.1-20230828-1build1) ...  
Setting up libsasl2-modules-db:amd64 (2.1.28+dfsg1-5ubuntu3) ...  
Setting up ca-certificates (20240203) ...  
debconf: unable to initialize frontend: Dialog  
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/perl5/D  
ebconf/FrontEnd/Dialog.pm line 79.)  
debconf: falling back to frontend: Readline  
Updating certificates in /etc/ssl/certs...  
146 added, 0 removed; done.  
Setting up perl (5.38.2-3.2build2) ...  
Setting up libgssapi-krb5-2:amd64 (1.20.1-6ubuntu2) ...  
Setting up libsasl2-2:amd64 (2.1.28+dfsg1-5ubuntu3) ...  
Setting up libssh-4:amd64 (0.10.6-2build2) ...  
Setting up libx11-6:amd64 (2:1.8.7-1build1) ...  
Setting up libldap2:amd64 (2.6.7+dfsg-1~exp1ubuntu8) ...  
Setting up libxml2:amd64 (2.10.3-1ubuntu1) ...  
Setting up libcurl3t64-gnutls:amd64 (8.5.0-2ubuntu10.1) ...  
Setting up libxext6:amd64 (2:1.3.4-1build2) ...  
Setting up liberror-perl (0.17029-2) ...  
Setting up git (1:2.43.0-1ubuntu7) ...  
Setting up xauth (1:1.1.2-1build1) ...  
Processing triggers for libc-bin (2.39-0ubuntu8.1) ...  
Processing triggers for ca-certificates (20240203) ...  
Updating certificates in /etc/ssl/certs...  
0 added, 0 removed; done.  
Running hooks in /etc/ca-certificates/update.d...  
done.  
root@a64e6017ba3b:/#
```

이미지 사용 방법- 직접 이미지를 만들어 사용

① 명령 프롬프트

- 명령 프롬프트에 docker pull ubuntu 입력

② 우분투 변경

- docker run -it --name [컨테이너 명]

ubuntu bash 입력하여 명령 프롬프트를

우분투로 변경

③ git 설치

- git 미 설치 시 apt update 입력 후

apt install git을 입력하여 git을 설치

실행중인 컨테이너를 이미지로 만들고, 두개의 독립된 환경을 구축하는 방법

1 exit 하여 우분투 탈출 후 명령 프롬프트로 이동 (우분투 환경이었을 경우)

2 docker commit [컨테이너 명] [새 이미지 이름]를 입력
예시) docker commit my-ubuntu egoing:ubuntu-git

두 개의 명령프롬프트 창을 열고 각각 아래 명령어를 입력

3 docker run [옵션] --name [컨테이너 이름] [이미지 이름] [command]
예시) docker run -it --name python egoing:ubuntu-git bash
예시) docker run -it --name nodejs egoing:ubuntu-git bash

두 개의 명령프롬프트 창에서 각각 아래를 입력

4 apt update && apt install [패키지 이름]
예시) apt update && apt install python3 (python은 지원X)
예시) apt update && apt install nodejs

5 독립된 환경 구축 완료

쿠버네티스 기본 명령어

```
C:\Users\User>kubectl apply -f C:\Users\User\Documents\kube_folder\apa000dep.yml
deployment.apps/apa000dep created
```

```
C:\Users\User>kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
apa000dep-749fc84c84-4sxww  1/1     Running   0          10s
apa000dep-749fc84c84-86dz7  1/1     Running   0          10s
apa000dep-749fc84c84-gvpr6  1/1     Running   0          10s
```

```

! apa000ser.yml ! apa000dep.yml X JS app.js •
C: > Users > User > Documents > kube_folder > ! apa000dep.yml
 1 apiVersion: apps/v1
 2 kind: Deployment
 3 metadata:
 4   name: apa000dep
 5 spec:
 6   selector:
 7     matchLabels:
 8       app: apa000kube
 9   replicas: 3
10   template:
11     metadata:
12       labels:
13         app: apa000kube
14     spec:
15       containers:
16         - name: apa000ex91
17           image: httpd
18         ports:
19           - containerPort: 80
20

```

- 클러스터 정보 확인

명령어: 'kubectl cluster-info'

- 매니페스트 파일 읽기

명령어: kubectl apply -f [파일경로]

예시) kubectl apply -f

C:\Users\User\Documents\kube_folder\apa000dep.yml

노드 리스트 확인

명령어: 'kubectl get nodes'

파드 리스트 확인

명령어: 'kubectl get pods'

오픈소스 SW 입문

감사합니다

Q&A

팀 J