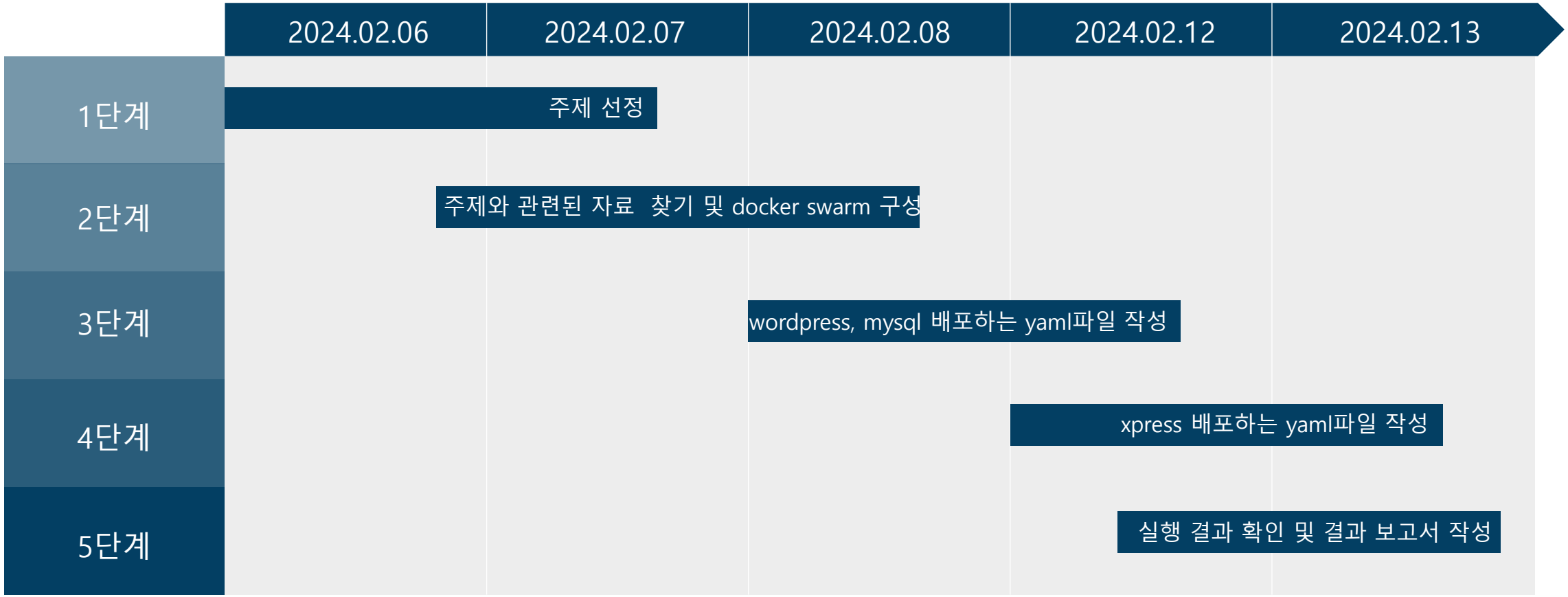
Docker swarm 프로젝트

2024.02.13

이원우

CONTENTS

프로젝트 일정


2024.02.06
2024.02.07
2024.02.08
2024.02.12
2024.02.13
1단계
2단계
3단계
4단계
5단계
주제 선정
주제와 관련된 자료 찾기 및 docker swarm 구성
wordpress, mysql 배포하는 yaml파일 작성
xpress 배포하는 yaml파일 작성
실행 결과 확인 및 결과 보고서 작성

01
Docker swarm 구성

노드 이름 및 ip구성

	ip	cpu	ram
manager	211.183.3.100	4	4
worker1	211.183.3.101	2	2
worker2	211.183.3.102	2	2
worker3	211.183.3.103	2	2

user1@manager:~$ docker swarm init --advertise-addr 211.183.3.100
211.183.3.100이 leader로써 docker swarm을 구성

user1@manager:~$ docker swarm join-token manager
To add a manager to this swarm, run the following command:

    docker swarm join –token SWMTKN-1-2pz8gtlcnptlgr7cjig12pgcxqafselu1dl17nh5k6fvzdinhq-2kxghzow8qqn9dlmlecbdjkw8 211.183.3.100:2377
위 토큰을 다른 노드에서 사용하면 manager로써 docker swarm에 참여시킬 수 있다
user1@manager:~$ docker swarm join-token worker
To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-2pz8gtlcnptlgr7cjig12pgcxqafselu1dl17nh5k6fvzdinhq-027g9ngp1wevsvgqwllm64lry 211.183.3.100:2377
위 토큰을 다른 노드에서 사용하면 worker로써 docker swarm에 참여시킬 수 있다

```
user1@manager:~$ docker node update --label-add zone=A --label-add app=web worker1
### worker1에 zone=A, app=web 이라는 label을 추가해 준다
user1@manager:~$ docker node update --label-add zone=A --label-add app=web worker2
### worker2에 zone=A, app=web 이라는 label을 추가해 준다
user1@manager:~$ docker node update --label-add zone=B --label-add app=db worker3
### worker1에 zone=B, app=db 라는 label을 추가해 준다

### 추가한 label을 확인할 수 있다
user1@manager:~$ docker node inspect worker1 --format="{{ .Spec }}"
-> {{ map[app:web zone:A]} worker active}
user1@manager:~$ docker node inspect worker2 --format="{{ .Spec }}"
-> {{ map[app:web zone:A]} worker active}
user1@manager:~$ docker node inspect worker3 --format="{{ .Spec }}"
-> {{ map[app:db zone:B]} worker active}
```

02

Manager 노드에 elk, kibana 설치

```
user1@manager:~$ docker pull docker.elastic.co/elasticsearch/elasticsearch:7.17.7
### docker로 elasticsearch pull해서 설치
user1@manager:~$ docker pull docker.elastic.co/kibana/kibana:7.17.7
### docker로 kibana pull해서 설치
user1@manager:~$ docker run -d -p 211.183.3.100:9200:9200 --name elasticsearch
docker.elastic.co/elasticsearch/elasticsearch:7.17.7
### docker image로 containe만들기
### -p옵션으로 포트지정 (9200:9200)
### -d옵션으로 컨테이너를 백그라운드에서 실행하고 출력을 현재 터미널에 표시하지 않는다
user1@manager:~$ docker run -d -p 211.183.3.100:5601:5601 --link elasticsearch:elasticsearch --
name kibana docker.elastic.co/kibana/kibana:7.17.7
### -p옵션으로 포트지정 (5601:5601)
### -d옵션으로 컨테이너를 백그라운드에서 실행하고 출력을 현재 터미널에 표시하지 않는다
```

container 확인
user1@manager:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES 29d637d36c11
docker.elastic.co/kibana/kibana:7.17.7 "/bin/tini -- /usr/l…" 4 minutes ago Up 3 minutes 5601/tcp
es_kibana_kibana.8lbwm5mi7t730dg987ygz9jj4.pt4508ht8x7fzup29k196ey1h 1132541655d7
docker.elastic.co/elasticsearch/elasticsearch:7.17.7 "/bin/tini -- /usr/l…" 4 minutes ago Up 4 minutes
9200/tcp, 9300/tcp es_kibana_es.8lbwm5mi7t730dg987ygz9jj4.x169z0sciq8z4f13kim7oz2te

es_kibana.yaml 파일

```
version: '3.7'
services:
  es:
    image:
docker.elastic.co/elasticsearch/elasticsearch:7.17.7
    environment:
      - node.name=single
      - cluster.name=standalone
      - discovery.type=single-node
    volumes:
      - data:/usr/share/elasticsearch/data
    ports:
      - 9200:9200
    networks:
      - es-bridge
    deploy:
      mode: global
      placement:
        constraints: [node.role == manager]
      restart_policy:
        condition: on-failure
        max_attempts: 3
```

es_kibana.yaml 파일

```
db:
    image: mysql:5.7
    ports:
      - "33061-33069:3306"
    deploy:
      mode: replicated
      replicas: 2
      placement:
        constraints: [ node.labels.zone == A ]
    environment:
      MYSQL_ROOT_PASSWORD: wordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress
      restart: always
      volumes:
        - db_data:/var/lib/mysql
      networks:
        - web_net

networks:
  web_net: {}

volumes:
  db_data: {}

  web_data: {}
```

es_kibana.yaml 파일 실행

만든 es_kibana.yaml파일 배포하기
user1@manager:~$ docker stack deploy -c=es_kibana.yaml es_kibana
Updating service es_kibana_kibana (id: ozbpn4jn8cp1pplqcz69b4cva)
Updating service es_kibana_es (id: b3aqi63qsoryzoa7yedrwplib)

chrome (외부 네트워크)으로 접속해서 확인해보기
http://211.183.3.100:5601

홈페이지가 접속되는 걸 확인할 수 있다

Home - Elastic

주의 요함 211.183.3.100:5601/app/home#/

elastic

Search Elastic

D Home

Welcome home

Enterprise Search

Create search experiences with a refined set of APIs and tools.

Observability

Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.

Security

Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.

Analytics

Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.

Get started by adding integrations

To start working with your data, use one of our many ingest options. Collect data from an app or service, or upload a file. If you're not ready to use your own data, add a sample data set.

Add integrations | Try sample data | Upload a file

Management

Dev Tools | Stack Management

03

Manager에서 worker1, worker2에 wordpress, mysql 배포하기

docker-compose.yaml 파일 작성

```
user1@manager:~/0212$
cat docker-compose.yaml

version: "3.8"

# 동일 기능을 제공하는 컨테이너들은 서비스로 grouping된다.
services:
  web:
    image: wordpress
    ports:
      - "8001-8009:80"
    deploy:
      mode: replicated
      replicas: 2
      placement:
        constraints: [ node.labels.zone == A ]
    restart: always
    volumes:
      - web_data:/var/www/html
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD:
wordpress
      WORDPRESS_DB_NAME: wordpress
    depends_on:
      - db

    networks:
      - web_net
```

docker-compose.yaml 파일 작성

```
db:
    image: mysql:5.7
    ports:
      - "33061-33069:3306"
    deploy:
      mode: replicated
      replicas: 2
      placement:
        constraints: [ node.labels.zone == A ]
    environment:
      MYSQL_ROOT_PASSWORD: wordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress
    restart: always
    volumes:
      - db_data:/var/lib/mysql
    networks:
      - web_net

networks:
  web_net: {}
volumes:
  db_data: {}

  web_data: {}
```

Docker-compose.yaml 파일 실행

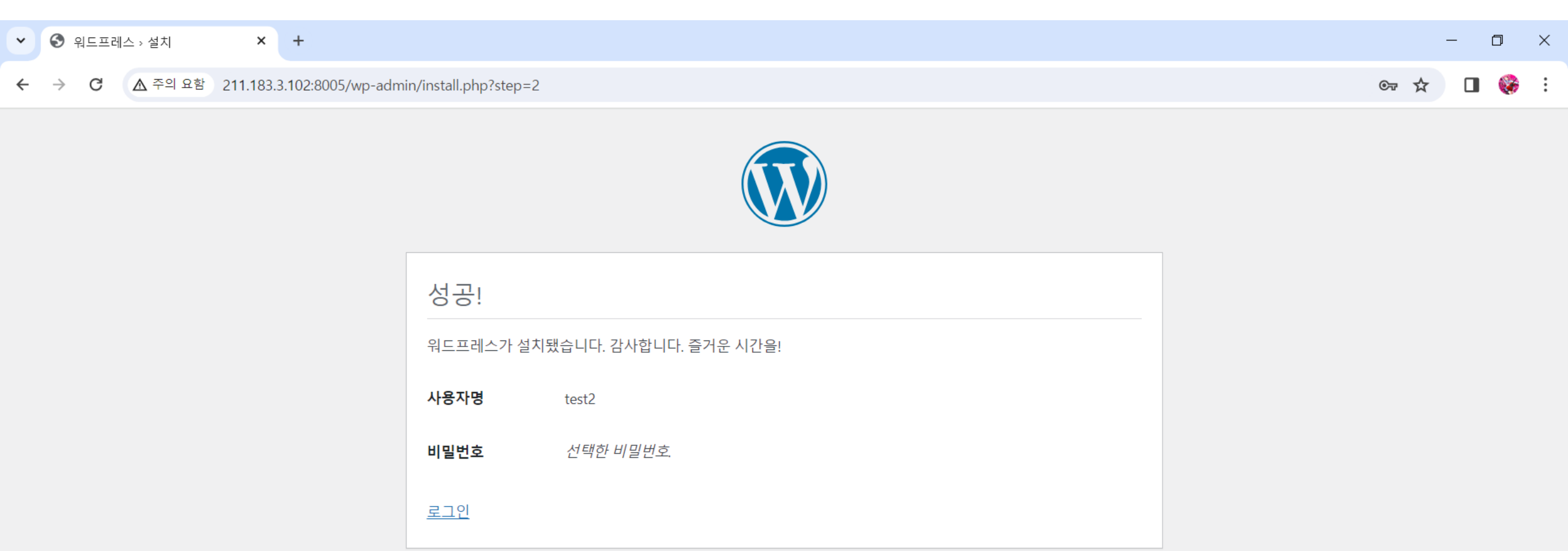
```
### docker-compose.yaml파일 실행
user1@manager:~/0212$ docker-compose up –d

### docker-compose.yaml파일 실행 결과 확인
user1@manager:~/0212$ docker stack ps web
ID             NAME         IMAGE              NODE      DESIRED STATE    CURRENT STATE
ERROR      PORTS
ez1xkyehj4jm    web_db.1     mysql:5.7          worker2    Running          Running 3 hours ago
nio9jphls3od    web_db.2     mysql:5.7          worker1    Running          Running 3 hours ago
ez3hq8f9pl47    web_web.1    wordpress:latest    worker1    Running          Running 3 hours ago
lgfwv34be42q    web_web.2    wordpress:latest    worker2    Running          Running 3 hours ago
### zone == A로 설정 된 worker1, worker2에만 설치된 모습을 확인할 수 있다
```

04

worker1, worker2 에서 wordpress, mariadb 확인해보기

Worker1에서 wordpress 정상 실행



Worker2에서 wordpress 정상 실행

test1

주의 요함 211.183.3.101:8001

test1 예제 페이지

A commitment to innovation and sustainability

Études is a pioneering firm that seamlessly merges creativity and functionality to redefine architectural excellence.

About us

test2

주의 요함 211.183.3.102:8005

test2 예제 페이지

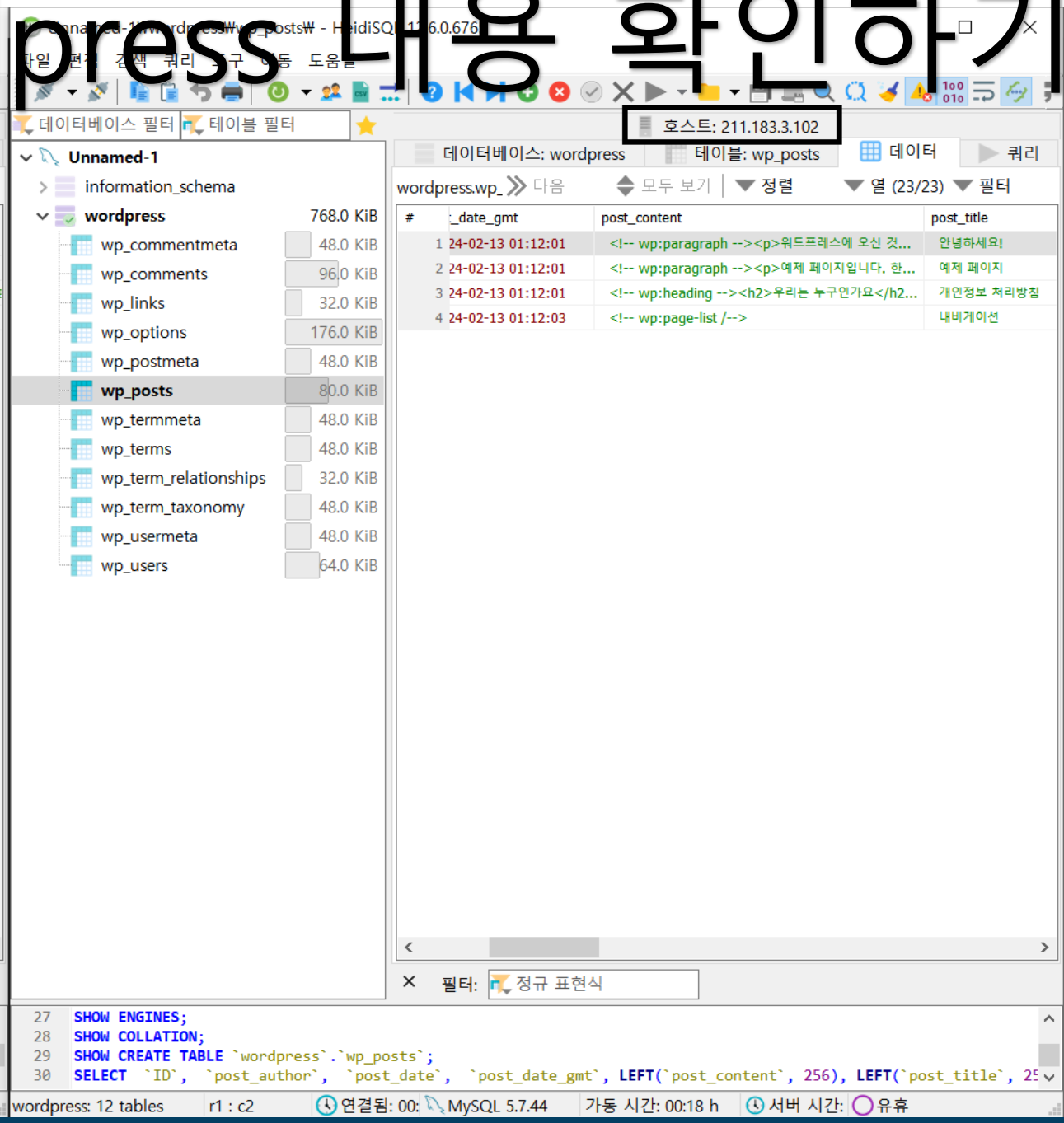
A commitment to innovation and sustainability

Études is a pioneering firm that seamlessly merges creativity and functionality to redefine architectural excellence.

About us

Worke1,2에서 wordpress 가입 및 페이지 화면

HeidiSQL에서 wordpress 내용 확인하기



05

이미지 만들어서 docker hub 업로드

(Label.app == db에 xpress engine 배포해보기)

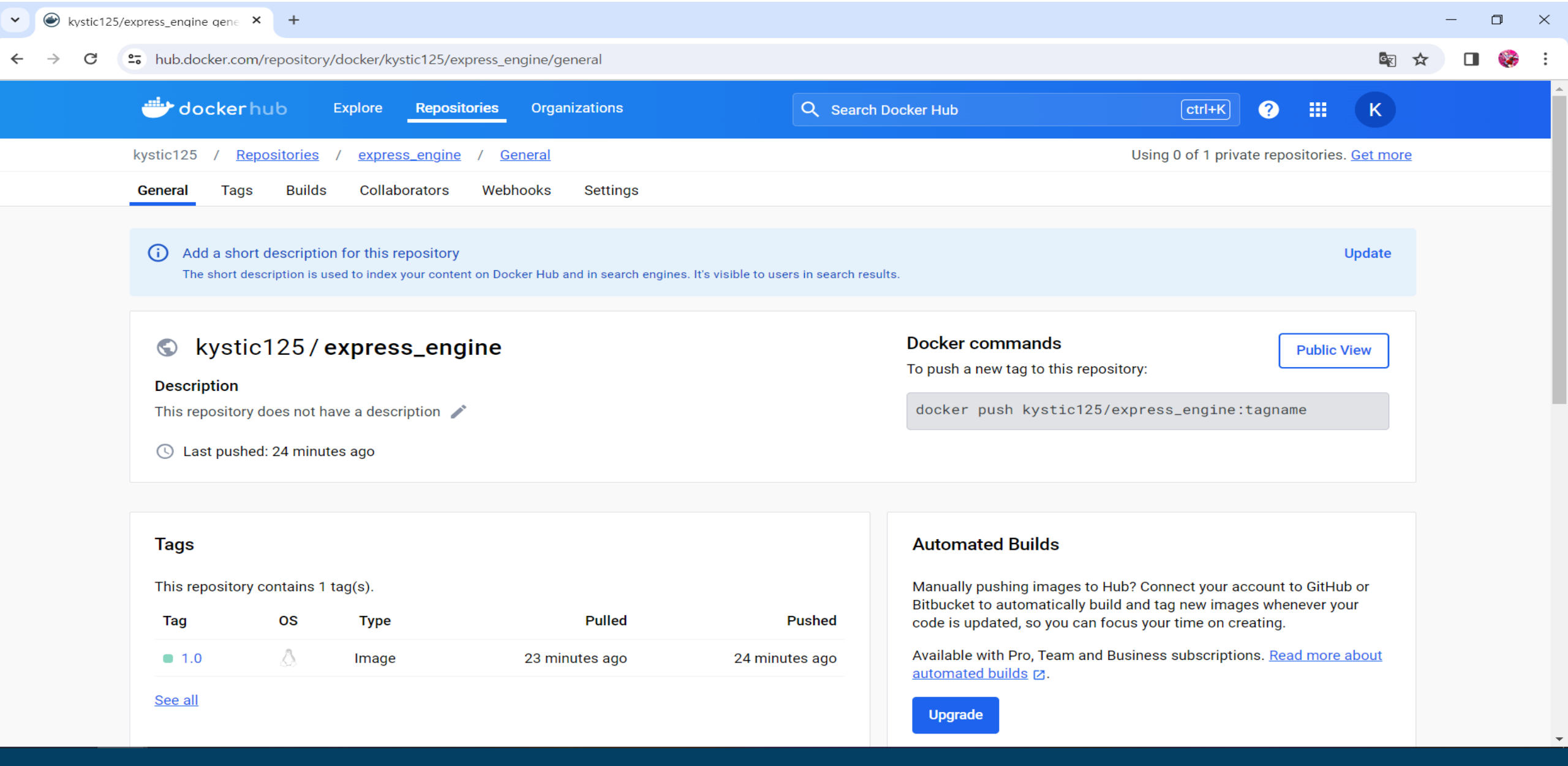
Dockerfile 작성

Dockerfile 만들기
user1@manager:~/0212$ vi Dockerfile
```
FROM centos:7
RUN yum clean all
RUN yum update -y
RUN yum -y install wget git httpd
RUN wget http://rpms.remirepo.net/enterprise/remi-release-7.rpm
RUN yum -y localinstall remi-release-7.rpm
RUN yum -y install epel-release yum-utils
RUN yum-config-manager --enable remi-php74
RUN yum -y install php php-fpm php-gd php-mysql php-xml
RUN git clone https://github.com/xpressengine/xe-core.git /var/www/html/xe
WORKDIR /var/www/html/xe
RUN mkdir files
WORKDIR /var/www/html
RUN chmod -R 707 xe
RUN chown -R apache:apache xe EXPOSE 80
CMD httpd -D FOREGROUND
```

docker image build하기
user1@manager:~/0212$ docker build -t kystic125/express_engine:1.0 .

docker hub에 이미지 업로드
user1@manager:~/0212$ docker push kystic125/express_engine:1.0

업로드 된 이미지 확인


kystic125/express_engine gene
hub.docker.com/repository/docker/kystic125/express_engine/general
docker hub
Explore
Repositories
Organizations
Search Docker Hub
ctrl+K
K
kystic125 / Repositories / express_engine / General
Using 0 of 1 private repositories. Get more
General
Tags
Builds
Collaborators
Webhooks
Settings
Add a short description for this repository
The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.
Update
kystic125 / express_engine
Description
This repository does not have a description
Last pushed: 24 minutes ago
Docker commands
To push a new tag to this repository:
Public View
docker push kystic125/express_engine:tagname
Tags
This repository contains 1 tag(s).
Tag
OS
Type
Pulled
Pushed
1.0
Image
23 minutes ago
24 minutes ago
See all
Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.
Available with Pro, Team and Business subscriptions. Read more about automated builds.
Upgrade

06

Label.zone == A에 대해서만 xpress engine 배포 및 확인해보기

(worker3만 label.zone == A인 상태)

!!! Worker3 !!!에서 이미지 가져오기

```
### docker hub에서 pull로 이미지 가져오기
user1@worker3:~$ docker pull kystic125/express_engine:1.0
### pull 한 이미지 확인하기
user1@worker3:~$ docker image ls
kystic125/express_engine                    1.0       d86438ab761d   45 minutes ago   2.26GB
```

배포를 위한 docker-compose1.yaml파일 작성

```
user1@manager:~/0212$
cat docker-compose1.yaml

version: '3.7'

services:
  xpress:
    image: kystic125/express_engine:1.0
    ports:
      - "8888:80"
    deploy:
      placement:
        constraints:
          - node.labels.app == db
```

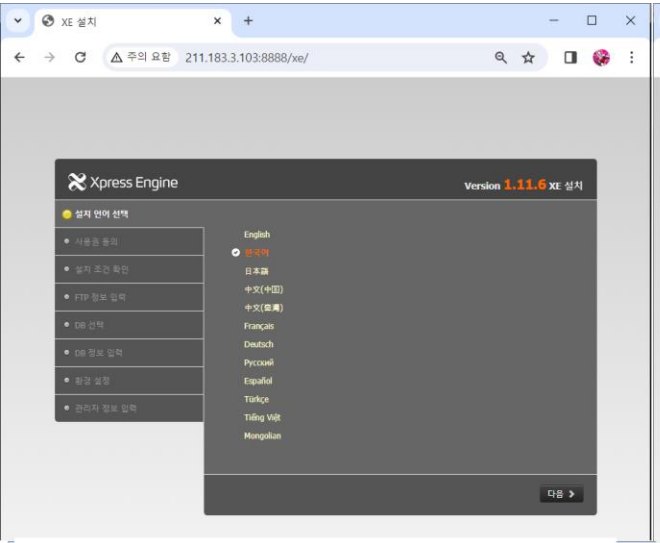
```
mysql:
  image: mysql:5.7
  environment:
    MYSQL_ROOT_PASSWORD: test123
    MYSQL_DATABASE: xe
  deploy:
    placement:
      constraints:
        - node.labels.app == db
```

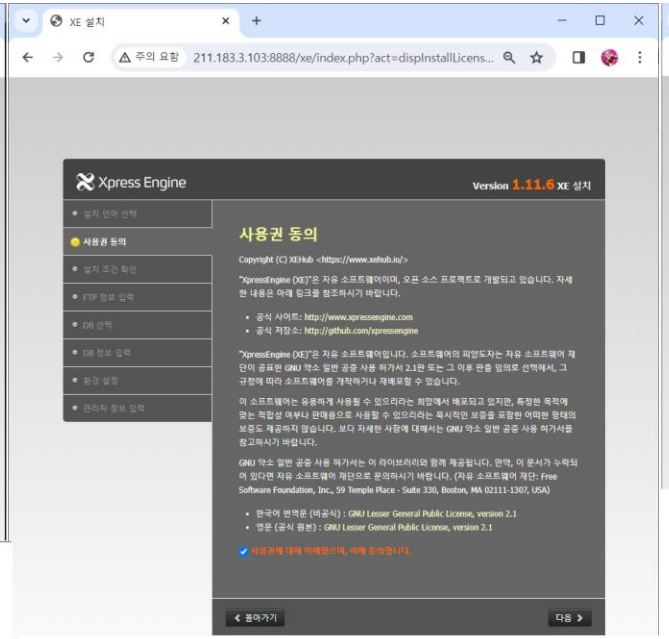
이번에는 db의 port를 열지 않음

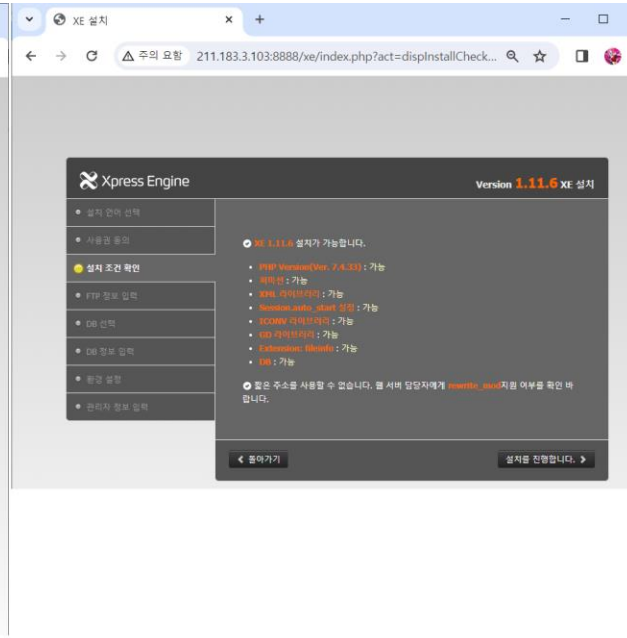
배포를 위한 docker-compose1.yaml파일 작성

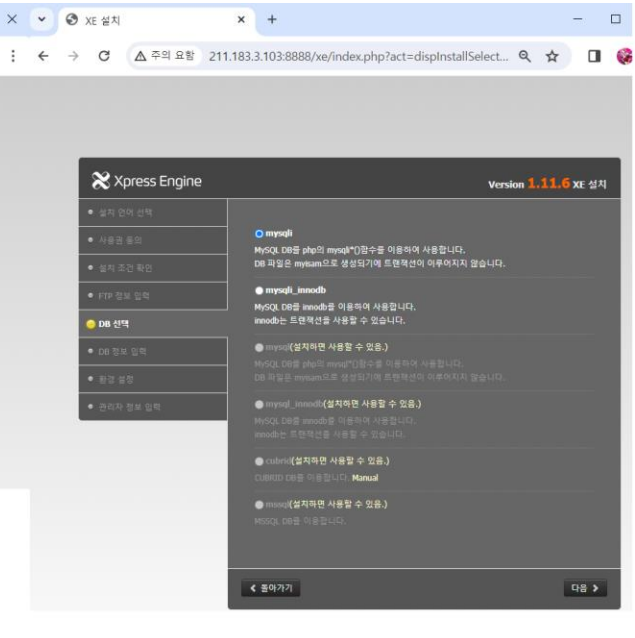
```
### docker-compose1.yaml파일을 이용하여 배포 명령하기
user1@manager:~/0212$ docker stack deploy -c docker-compose1.yaml xe
### 배포 결과 확인하기
user1@manager:~/0212$ docker stack ps xe
ID             NAME          IMAGE                          NODE      DESIRED STATE  CURRENT STATE
ERROR      PORTS
Wpjkvqty69wa   xe_mysql.1    mysql:5.7                      worker3   Running        Running 2
minutes ago
0n0jm50af1er   xe_xpress.1   kystic125/express_engine:1.0   worker3   Running        Running 2
minutes ago
```

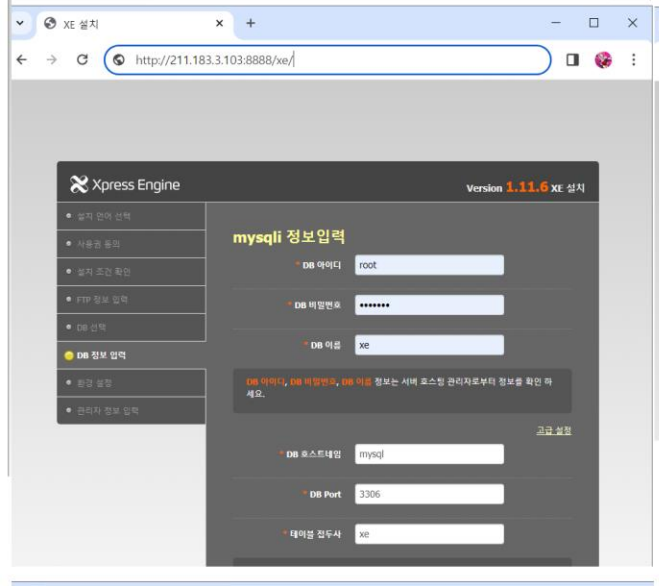
다음 페이지
: 211.183.3.103:8888/xe에 접속해서 확인해보기

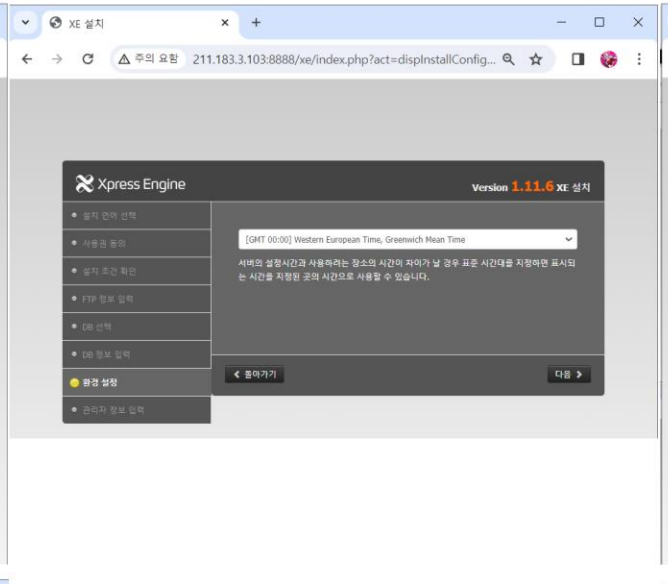

XE 설치
211.183.3.103:8888/xe/
Xpress Engine
Version 1.11.6 XE 설치
설치 언어 선택
사용권 동의
설치 조건 확인
FTP 정보 입력
DB 선택
DB 정보 입력
환경 설정
관리자 정보 입력
English
한국어
日本語
中文(中国)
中文(臺灣)
Français
Deutsch
Русский
Español
Türkçe
Tiếng Việt
Mongolian
다음 ›

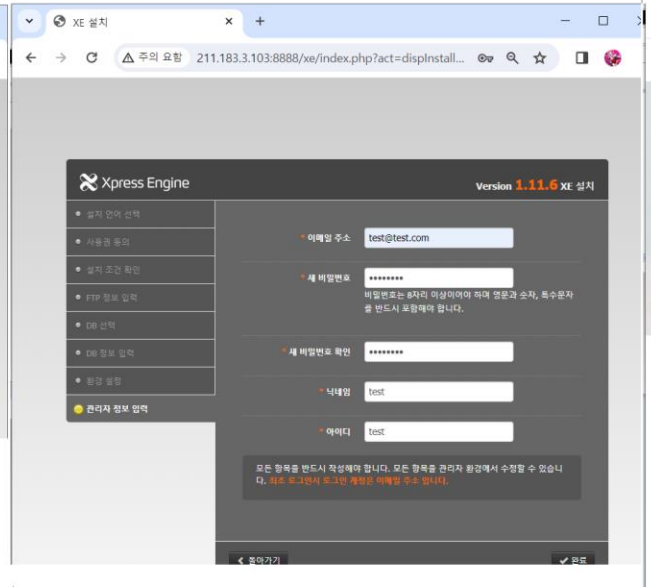

XE 설치
211.183.3.103:8888/xe/index.php?act=dispInstallLicens...
Xpress Engine
Version 1.11.6 XE 설치
사용권 동의
Copyright (C) XEHub <https://www.xehub.io/>
"XpressEngine (XE)"은 자유 소프트웨어이며, 오픈 소스 프로젝트로 개발되고 있습니다. 자세한 내용은 아래 링크를 참조하시기 바랍니다.
공식 사이트: http://www.xpressengine.com
공식 저장소: http://github.com/xpressengine
"XpressEngine (XE)"은 자유 소프트웨어입니다. 소프트웨어의 피양도자는 자유 소프트웨어 재단이 공표한 GNU 약소 일반 공중 사용 허가서 2.1판 또는 그 이후 판을 임의로 선택해서, 그 규정에 따라 소프트웨어를 개작하거나 재배포할 수 있습니다.
이 소프트웨어는 유용하게 사용될 수 있으리라는 희망에서 배포되고 있지만, 특정한 목적에 맞는 적합성 여부나 판매용으로 사용할 수 있으리라는 묵시적인 보증을 포함한 어떠한 형태의 보증도 제공하지 않습니다. 보다 자세한 사항에 대해서는 GNU 약소 일반 공중 사용 허가서를 참고하시기 바랍니다.
GNU 약소 일반 공중 사용 허가서는 이 라이브러리와 함께 제공됩니다. 만약, 이 문서가 누락되어 있다면 자유 소프트웨어 재단으로 문의하시기 바랍니다. (자유 소프트웨어 재단: Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA)
한국어 번역문 (비공식) : GNU Lesser General Public License, version 2.1
영문 (공식 원본) : GNU Lesser General Public License, version 2.1
사용권에 대해 이해했으며, 이에 동의합니다.
돌아가기
다음 ›

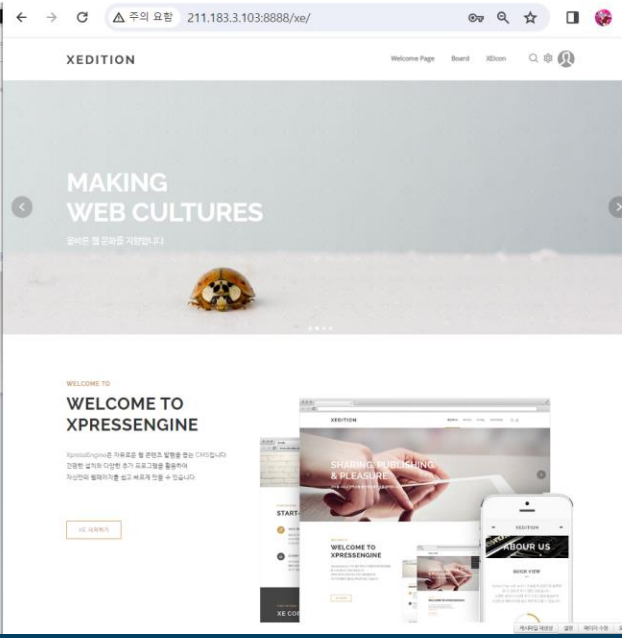

XE 설치
211.183.3.103:8888/xe/index.php?act=dispInstallCheck...
Xpress Engine
Version 1.11.6 XE 설치
XE 1.11.6 설치가 가능합니다.
PHP Version(Ver. 7.4.33) : 가능
파이선 : 가능
XML 라이브러리 : 가능
Session.auto_start 설정 : 가능
ICONV 라이브러리 : 가능
GD 라이브러리 : 가능
Extension: fileinfo : 가능
DB : 가능
짧은 주소를 사용할 수 없습니다. 웹 서버 담당자에게 mod_rewrite 지원 여부를 확인 바랍니다.
돌아가기
설치를 진행합니다. ›


XE 설치
211.183.3.103:8888/xe/index.php?act=dispInstallSelect...
Xpress Engine
Version 1.11.6 XE 설치
DB 선택
mysqli
MySQL DB를 php의 mysqli"()함수를 이용하여 사용합니다.
DB 파일은 myisam으로 생성되기에 트랜잭션이 이루어지지 않습니다.
mysqli_innodb
MySQL DB를 innodb를 이용하여 사용합니다.
innodb는 트랜잭션을 사용할 수 있습니다.
mysql(설치하면 사용할 수 있음.)
mysqli_innodb(설치하면 사용할 수 있음.)
cubrid(설치하면 사용할 수 있음.)
CUBRID DB를 이용합니다. Manual
mssql(설치하면 사용할 수 있음.)
MSSQL DB를 이용합니다.
돌아가기
다음 ›


XE 설치
http://211.183.3.103:8888/xe/
Xpress Engine
Version 1.11.6 XE 설치
mysqli 정보입력
DB 아이디 root
DB 비밀번호 ••••••
DB 이름 xe
DB 아이디, DB 비밀번호, DB 이름 정보는 서버 호스팅 관리자로부터 정보를 확인 하세요.
고급 설정
DB 호스트네임 mysql
DB Port 3306
테이블 접두사 xe


XE 설치
211.183.3.103:8888/xe/index.php?act=dispInstallConfig...
Xpress Engine
Version 1.11.6 XE 설치
[GMT 00:00] Western European Time, Greenwich Mean Time
서버의 설정시간과 사용하려는 장소의 시간이 차이가 날 경우 표준 시간대를 지정하면 표시되는 시간을 지정한 곳의 시간으로 사용할 수 있습니다.
돌아가기
다음 ›


XE 설치
211.183.3.103:8888/xe/index.php?act=dispInstall...
Xpress Engine
Version 1.11.6 XE 설치
이메일 주소 test@test.com
새 비밀번호 ••••••••
비밀번호는 8자리 이상이어야 하며 영문과 숫자, 특수문자를 반드시 포함해야 합니다.
새 비밀번호 확인 ••••••••
닉네임 test
아이디 test
모든 항목을 반드시 작성해야 합니다. 모든 항목을 관리자 환경에서 수정할 수 있습니다. 최초 로그인시 로그인 계정은 이메일 주소 입니다.
돌아가기
완료


211.183.3.103:8888/xe/
XEDITION
MAKING
WEB CULTURES
WELCOME TO
WELCOME TO
XPRESSENGINE
SHARING, PUBLISHING
& PLEASURE
ABOUR US

docker-compose.yaml 파일 작성

```
db:
  image: mysql:5.7
  ports:
    - "33061-33069:3306"
  deploy:
    mode: replicated
    replicas: 2
    placement:
      constraints: [ node.labels.zone == A ]
  environment:
    MYSQL_ROOT_PASSWORD: wordpress
    MYSQL_DATABASE: wordpress
    MYSQL_USER: wordpress
    MYSQL_PASSWORD: wordpress
  restart: always
  volumes:
    - db_data:/var/lib/mysql
  networks:
    - web_net

networks:
  web_net: {}
volumes:
  db_data: {}

  web_data: {}
```

docker-compose.yaml 파일 작성

```
db:
  image: mysql:5.7
  ports:
    - "33061-33069:3306"
  deploy:
    mode: replicated
    replicas: 2
    placement:
      constraints: [ node.labels.zone == A ]
  environment:
    MYSQL_ROOT_PASSWORD: wordpress
    MYSQL_DATABASE: wordpress
    MYSQL_USER: wordpress
    MYSQL_PASSWORD: wordpress
  restart: always
  volumes:
    - db_data:/var/lib/mysql
  networks:
    - web_net

networks:
  web_net: {}
volumes:
  db_data: {}

  web_data: {}
```

07

Metricbeat를 이용한 데이터 수집 및 manager에서 시각화

metricbeat 설치 (Manager, worker1, 2에서 전부 실행)

metricbeat 설치
user1@manager:~$ curl –L –O https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-7.17.7-amd64.deb
metricbeat 압축 해제
user1@manager:~$ sudo dpkg –I metricbeat-7.17.7-amd64.deb
metricbeat.yml 파일 수정
user1@manager:~$ sudo vi /etc/metricbeat/metricbeat.yml
""" 아래와 같이 수정
61 setup.kibana:
62 host: "211.183.3.100:5601"
93 output.elasticsearch:
94 # Array of hosts to connect to.
95 hosts: ["http://211.183.3.100:9200"]
96 username: "elastic"
97 password: "test123"
"""

metricbeat 재시작
user1@manager:~$ sudo metricbeat setup
user1@manager:~$ sudo service metricbeat stop
user1@manager:~$ sudo service metricbeat start

metricbeat module 활성화
user1@manager:~$ sudo metricbeat modules enable docker

/etc/metricbeat/modules.d/docker.yml 파일 수정

```
# Module: docker
# Docs:
https://www.elastic.co/guide/en/beats/metricbeat/7.17/m
etricbeat-module-docker.html

- module: docker
  metricsets:
    - container
    - cpu
    - diskio
    - event
    - healthcheck
    - info
    - memory
    - network
    - network_summary
  period: 10s
```

[Metricbeat Docker] Overview

211.183.3.100:5601/app/dashboards#/view/AV4REOpp5NkDleZmzKkE-ecs?_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from:now-15m,to:now))

elastic

Search Elastic

Dashboard / [Metricbeat Docker] Overview ECS

Full screen | Share | Clone | Edit

Search | KQL | Last 15 minutes | Show dates | Refresh

+ Add filter

시각화 확인하기

Docker containers [Metricbeat Docker] ECS

Export

Name	CPU usage (%)	DiskIO	Mem (%)	Mem RSS	Number of Containers
0212-web-1	0%	0	0.4%	11.9MB	1
0212-web-2	0%	0.36	0.4%	11.8MB	1
0212-db-1	0.2%	3.364	2.3%	174.2MB	1
es_kibana_es.qmswoae...	79.4%	65.433	46.7%	3.5GB	1
es_kibana_kibana.qms...	98.9%	9.657	4.6%	345.7MB	1
	178.5%	**78.814**	**54.4%**	**4.1GB**	**5**

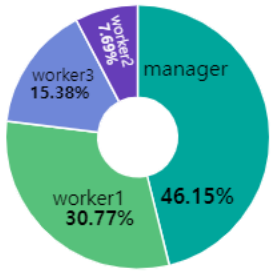
Number of Containers [Metricbeat Docker] ECS

6 Running

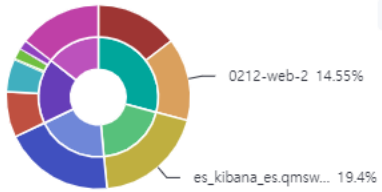
0 Paused

5 Stopped

Docker containers per host [Metricbeat Dock...



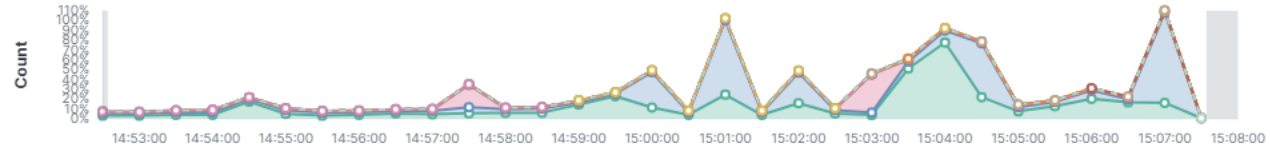
Docker images and names [Metricbeat Docker] ECS



wordpress | docker.elastic.co/elastics... | docker.elastic.co/kibana/k... | mysql:5.7@sha256:4bc6b...

CPU usage [Metricbeat Docker] ECS

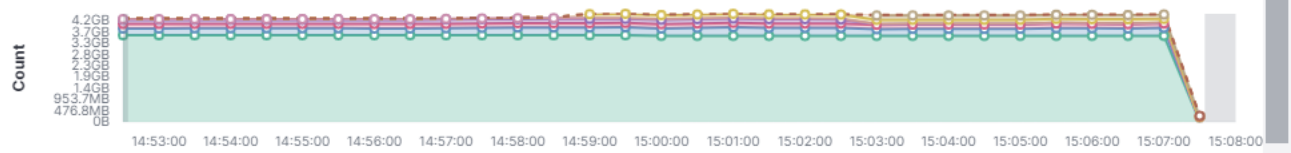
web_web.2.wwwnom5u2u... | web_web.1.z5n2tri1figw5q... | web_db.2.vspoawj4q6oz0... | web_db.2.561ep06y87wn4... | web_db.1.rpjr5u95hbp03i4... | xe_mysql.1.wpjkvqty69waj... | 0212-web-1 | 0212-db-1



@timestamp per 30 seconds

Memory usage [Metricbeat Docker] ECS

web_web.1.z5n2tri1figw5q... | web_web.2.wwwnom5u2u... | web_db.1.rpjr5u95hbp03i4... | xe_mysql.1.wpjkvqty69waj... | 0212-web-2 | 0212-db-1 | kibana | es_kibana_kibana.qmswoa...



@timestamp per 30 seconds

Network IO [Metricbeat Docker] ECS

web_web.1.z5n2tri1figw5q... | web_db.2.vspoawj4q6oz0... | web_db.1.rpjr5u95hbp03i4... | web_web.2.wwwnom5u2u... | kibana: OUT bytes | es_kibana_kibana.qmswoa... | es_kibana_es.qmswoaexfr... | 0212-web-2: OUT bytes

0212-web-1: OUT bytes | 0212-db-1: OUT bytes | web_web.1.z5n2tri1figw5q... | web_db.2.vspoawj4q6oz0... | web_db.1.rpjr5u95hbp03i4... | web_web.2.wwwnom5u2u... | kibana: IN bytes | es_kibana_kibana.qmswoa...

08

프로젝트 진행 후기 및 향후 보완점

진행 후기

- Docker Swarm을 사용하여 여러 서비스를 관리하고 배포하는 과정을 통해
  컨테이너 오케스트레이션에 대한 이해력을 향상시킬 수 있었습니다.

- ELK 스택 및 Metricbeat를 사용하여 로그와 메트릭 정보를 수집하고 시각화를 진행하면서
  수집된 데이터를 시각화를 진행하지 않고 볼 때와 비교하여 시각화의 필요성에 대해 느낄 수 있었습니다.
  향후 프로메테우스 및 그라파나와 같은 도구도 활용해 볼 예정입니다.

- Docker hub를 사용하면 컨테이너 이미지를 손쉽게 저장하고 공유할 수 있다는 것을 배웠습니다.
  이것을 활용하면 여러 프로젝트에서 사용할 수 있는 이미지를 효율적으로 관리할 수 있을 것 같다고 생각했습니다.

향후 보완점

- 단일 Manager 노드는 SPOF를 가지고 있습니다. 이 노드가 다운되면 전체 Swarm 클러스터의 가용성이 저하되므로 다른 노드를 manager로 join시켜 고가용성을 확보해야 합니다.
- Wordpress의 DB replica가 제공되지 않으므로 원본 DB에 데이터가 손실될 경우, 백업이 없다면 중요 정보를 복구하기 어려울 수 있습니다.
- 처음 목표에선 manager가 git에 파일을 push 하면 worker노드에서 해당 파일을 자동으로 pull로 가져오도록 github action을 구성하고 싶었으나 구현하지 못한 것이 아쉬움으로 남습니다.
  향후 jenkins를 이용하여 해당 기능을 보완해 볼 예정입니다.