도커컴포즈 사용법

- description : 도커컴포즈 사용법
- author : 치킨5분컷
- email : hgkim@repia.com
- lastupdate : 2022-11-22 Thu

사전에 설치가 필요한 프로그램

- 도커
- 리눅스 환경 프로그램

도커컴포즈란?

- 도커 컴포즈는 하나의 설정파일로 여러 컨테이너를 운용하게 할 수 있게 해주는 도구 중 하나.
- 여러 개의 컨테이너를 코드(YAML 파일)를 통해서 정의하고 실행하는 역할을 하며, 설정 파일인 `yaml`을 기반으로 동작.
- 재사용 가능

YAML (YAML Ain’t Markup Language)란

- 구조화 데이터나 오브젝트를 문자열로 변환하기 위한 데이터 형식의 한 종류.
- Space를 사용한 들여쓰기를 통해 데이터 계층을 표시.

★ 주의사항 - tab키 절대 사용 금지. 들여쓰기의 크기가 반드시 동일 해야 하기때문. 어길시 오류발생.

docker-compose.yml 특징

1. `YAML` 파일은 도커컴포즈에 관한 설정을 담음.
2. 확장자로 `.yml`, `.yaml`의 형식을 사용.
3. **key / value** 구조로 구성.
4. 대소문자를 구분.
5. 콜론(:) 뒤에는 한 칸을 띄고 사용. ex)`image: centos:7.9.2009`
6. 대시(-)로 복수의 값을 나열.
7. 주석 표기는 해시(#)를 이용.

★ 주의사항 - 콜론(:) 뒤에 단어 붙여쓰기 금지. 인식 불가로 인한 오류발생.

docker compose 파일의 구조

```
=====Top level 4가지(들여쓰기 금지)=====
```

```
version: () - 지원 버전을 확인
services: (docker container) - 컨테이너의 설정을 확인.
volumes: (docker volume) - 도커 볼륨에 대한 설명.
networks: (docker network) - 도커 네트워크에 대한 설명.
```

이렇게 크게 4가지를 잡고 도커컴포즈 파일을 생성.

- 도커 버전에 따라 지원 가능 버전이 달라지므로 확인 필요. 지원 버전 확인 사이트

compose file 버전별 차이

사용할려면 `docker-compose.yml` 상단에 사용할 compose file format의 버전을 명시.

버전별 특징

1. Version 1에서는 버저닝을 생략.
    - networking이 지원되지 않음
    - 모든 컨테이너는 기본 bridge 네트워크에 배치되고 해당 IP 주소의 다른 모든 컨테이너에서 연결 가능
    - 컨테이너 간 검색을 활성화하려면 links를 사용
2. Version 2부터 마이너 버전(2.x)까지 설정해야 함. (생략 시 2.0으로 적용된다.)
    - yml 문서에 버전명을 마이너 버전까지 작성 ex) 2.1
    - 모든 각각의 서비스는 **services 아래에 다중 컨테이너로 구성**
    - volumes에 Named volumes 생성 가능
    - networks에 Network 생성 가능
    - 기본적으로 **모든 컨테이너**는 **애플리케이션 전체의 기본 네트워크에 연결**되며 서비스 이름과 동일한 호스트 이름에서 검색 가능하다. 이것은 링크가 거의 불필요하다는 것을 의미한다.
3. Version 3 은 도커 스웜과 같이 사용되도록 디자인됨.

Docker Compose 명령어

- **docker-compose.yaml 파일이 위치한 디렉토리 에서 실행 (반드시)**
- 특정 파일의 위치를 정의.

```
1. docker-compose up                          # 서비스와 관련된 컨테이너 생성 그리고 시작
2. docker-compose down                        # 서비스와 관련된 컨테이너 종료 후 제거
3. docker-compose start/stop/restart          # 서비스 관련 컨테이너 시작 / 종료 / 재시작
4. docker-compose kill                        # 실행 중인 컨테이너에 SIGKH 시그널을 통해서 강제 종료
5. docker-compose pause/ unpause              # 컨테이너 정지 /재가동
```

```
6. --build                                    # 컨테이너를 생성하기 전에 이미지를 빌드한다.
```

Docker Compose 명령어 옵션

-d	백그라운드 실행
–no-deps	링크 서비스 실행하지 않음
–build	이미지 빌드
-t	타임아웃을 지정(기본 10초)

Docker Compose.yml 파일 내 사용가능한 옵션

구분	내용	명령어 순위
version :	버전 지정	최상위
services :	서비스 정의	최상위
volumes :	컨테이너에 볼륨 Mount	최상위
build :	빌드 환경 설정	service의 하위
context :	Dockerfile 이 있는 디렉토리 지정	build의 하위
dockerfile :	Dockerfile의 이름이 다를경우 설정	build의 하위
image :	이미지 지정	build의 하위
command :	container 에서 작동하는 명령	service의 하위
entrypoint :	container 에서 작동하는 명령	service의 하위
environment :	컨테이너 안 환경변수 설정	service의 하위
container_name :	컨테이너 이름 설정	service의 하위
links :	다른 컨테이너와 연결	service의 하위
ports :	Host 에 컨테이너가 공개하는 포트지정	service의 하위
expose :	컨테이너끼리 공개하는 포트 지정	service의 하위

도커컴포즈를 통한 컨테이너 생성 방법

```
1. hgkim@LAPTOP-266L93S4:/usr/bin$ sudo mkdir DockerCompose
# DockerCompose디렉토리 만들어 주기.
```

```
2. hgkim@LAPTOP-266L93S4:/usr/bin/DockerCompose$ sudo vi Dockerfile
# 도커파일의 설정 파일 작성.
```

```
FROM 만들 이미지 파일 (이미지이름:버전)
# FROM 필수.

추가적으로 필요한 명령어
```

```
3. hgkim@LAPTOP-266L93S4:/usr/bin/DockerCompose$ sudo vi docker-compose.yml
# 도커컴포즈 설정 파일 작성.
```

```
### docker-compose.yml
version: "3.9"                    #버전
services:                         #컨테이너 설정
```

```
apserver:                       #만들 서비스 이름
  build:                        #변경사항이 많아 Dockerfile을 빌드해야할 경우 이용
   context: .                   #Dockerfile의 위치. '.'은 현재 디렉토리
  ports:                        #Port포워딩할 포드범위
  - '22:22'
  - '7100-7110:7100-7110'
  - '80:80'
  - '8080:8080'
  user: root
  privileged: true              #기본적으로는 Unprivileged모드로 실행되며, 이 모드에서는 시스템 주요 자원에 접근할 수 있는 권한이 부족하여, 특정시스템 자원에 접근하기 위해 모드 변경.
  container_name: rsa_server #생성될 컨테이너 이름.
```

```
4. docker-compose up -d --build                 # 도커컴포즈 실행시 도커파일을 빌드 후 백그라운드 실행)
5. docker exec -it 컨테이너명 /bin/bash         # 컨테이너 실행.
```

도커컴포즈와 도커파일을 통한 응용 방법

- 검색솔루션에 사용할 재사용 가능한 적절한 OS 컨테이너를 만들 수 있음.

치킨5분컷, 도커컴포즈, 도커컴포즈 명령어, 도커컴포즈 컨테이너

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:3party:docker:%EB%8F%84%EC%BB%A4%EC%BB%B4%ED%8F%AC%EC%A6%88_%EC%82%AC%EC%9A%A9%EB%B2%95&rev=1669085408

Last update: **2022/11/22 11:50**