

[클라우드 컴퓨팅 기말 준비 정리]

1. Docker 개념 핵심 정리

(1) 컨테이너란?

- ☞ 컨테이너 = 독립된 실행 환경(서버 같은 역할)
- ☞ 하나의 서버에 여러 컨테이너를 띄워 효율적으로 서비스 운영
- ☞ 가상머신(VM)보다 훨씬 가볍고 빠름

(2) Docker란?

- ☞ 컨테이너를 만들고 실행하고 관리하는 프로그램
- ☞ 애플리케이션을 이미지(Image) 로 패키징 → 어디서든 동일하게 실행

개념	설명
이미지(Image)	실행 환경 + 프로그램이 담긴 설계도
컨테이너(Container)	이미지를 실행한 실제 인스턴스
Dockerfile	이미지를 자동 빌드하는 스크립트
Docker Hub	이미지 저장소 (GitHub의 Docker 버전)

(3) 컨테이너 라이프사이클

- ☞ 생성(create) → 실행(start/run) → 일시정지(pause) → 중지(stop) → 삭제(rm)

2. HTML + CSS로 홈페이지 제작

(1) 구성 페이지 (기본 HTML/CSS)

- ▶ index.html (홈)
- ▶ about.html (자기소개)
- ▶ projects.html (프로젝트 소개)
- ▶ board.html → 이후 PHP 게시판으로 대체
- ▶ style.css : 전체 레이아웃/색상 정의

(2) Apache2 + PHP + MySQL 환경 구성

- ▶ Apache 설치

```
sudo apt install -y apache2
```

- ▶ PHP + MySQL 연동 모듈

```
sudo apt install -y php libapache2-mod-php php-mysql
```

- ▶ MySQL 설치

```
sudo apt install -y mysql-server
```

- ▶ MySQL 접속

```
sudo mysql -u root -p
```

(3) MySQL + PHP 게시판(BOARD) 연동

▶ DB 구조

```
-- DB와 전용 계정 만들기 (원하는 이름으로 변경 가능)
CREATE DATABASE mysite CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE USER 'mysiteuser'@'localhost' IDENTIFIED BY 'StrongPassw0rd!';
GRANT ALL PRIVILEGES ON mysite.* TO 'mysiteuser'@'localhost';
FLUSH PRIVILEGES;

-- 게시판 테이블
USE mysite;
CREATE TABLE posts ( id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(50)
NOT NULL, title VARCHAR(200) NOT NULL, content TEXT NOT NULL, created_at
TIMESTAMP DEFAULT CURRENT_TIMESTAMP) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4;
```

(4) 게시판 정상 동작 테스트 방법

http://localhost/my_website/board.php

3. Docker 기반 배포 (로컬 환경)

(1) 전체구조

```
my_website_docker/
├── docker-compose.yml
├── .env
├── web/
│   ├── Dockerfile
│   └── php.ini
├── db/
│   └── init.sql
└── app/                                ← 실제 웹 소스(HTML/CSS/PHP)
    ├── index.html
    ├── about.html
    ├── projects.html
    ├── style.css
    ├── board.php
    ├── write.php
    ├── save.php
    ├── view.php
    ├── delete.php
    └── db.php
```

(2) 각 파일의 용도 정리해보기

4. 내용 정리

- (1) 컨테이너는 VM보다 가볍고, 애플리케이션 실행환경을 표준화.
- (2) 이미지 = 설계도 / 컨테이너 = 실행본, Dockerfile로 자동 빌드.
- (3) HTML/CSS + PHP로 만든 홈페이지를 PDO 기반으로 MySQL과 연결.
- (4) 게시판 CRUD 구현을 위해 6개 PHP 파일 구조가 핵심.
- (5) Docker Compose로 web + db 두 개의 서비스를 분리하여 운영.
- (6) Cloudtype은 docker-compose를 사용하지 않고 Dockerfile + app 폴더만 사용.
- (7) 배포 전 반드시 DB 연결(host/user/pass) 와 php.ini timezone 설정이 필요.

5. 용어

(1) 이미지(Image)

♣ 컨테이너를 생성하기 위한 실행 설계도

♣ OS + 라이브러리 + 실행환경 + 앱 코드가 포함됨

예: php:8.2-apache, mysql:8.0

(2) 컨테이너(Container)

♣ 이미지를 실제 실행한 독립된 실행 단위

♣ 서버처럼 동작 (애플리케이션 1개 = 컨테이너 1개 권장)

(3) Dockerfile

♣ 이미지를 자동으로 만들기 위한 명령어 스크립트

예: PHP 확장 설치, Apache 설정, php.ini 복사 등

(4) 레지스트리(Registry) / Docker Hub

♣ 이미지를 업로드·다운로드하는 저장소

♣ GitHub의 Docker 버전처럼 이해하면 편함

(5) 볼륨(Volume)

♣ 데이터 영속화(지속 저장)를 위해 컨테이너 외부에 만드는 저장공간

♣ MySQL 데이터 보존에 필수

(6) 네트워크(Network)

♣ 컨테이너끼리 통신을 가능하게 하는 가상 네트워크

♣ Docker Compose에서는 기본적으로 동일 네트워크에 배치됨

(7) 서비스(Service)

♣ Docker Compose에서 정의하는 컨테이너 단위 설정 블록

예: web, db 두 개의 서비스로 구성

(8) docker-compose.yml

♣ 여러 컨테이너(web, db 등)를 한 번에 관리하는 설정 파일

♣ 포트, 볼륨, 환경변수, 의존성, 빌드 설정을 모두 포함

(9) 환경 변수(.env)

- ♣ 여러 컨테이너가 사용할 설정 값(DB 계정, 패스워드 등)을 한 파일에 저장
- ♣ Compose 파일에서 자동 로드됨

(10) depends_on / healthcheck

- ♣ 컨테이너 실행 순서를 조정하는 설정
- ♣ DB가 정상 구동된 뒤 WEB을 실행하도록 제어
- ♣ 웹-DB 종속 구조를 안정적으로 유지하는 핵심 기능

6. 객관식 문제

(1) 도커(Docker)의 가장 핵심적인 목적은 무엇인가?

- ① 가상머신을 완전히 대체하는 OS 설치 도구
- ② 프로그램을 컨테이너라는 독립 실행 환경에서 동일하게 실행하도록 하는 것
- ③ 웹 서버만 실행 가능한 전용 플랫폼
- ④ 데이터베이스 백업 자동화 도구

(2) '이미지(Image)'의 정의로 가장 알맞은 것은?

- ① 실행 중인 컨테이너
- ② 프로그램 실행 결과 로그
- ③ 실행 환경과 프로그램이 포함된 설계도
- ④ 네트워크 설정 파일

(3) '컨테이너(Container)'의 설명 중 옳은 것은?

- ① 저장소에 올려두는 프로그램 패키지
- ② 이미지를 실행한 실제 인스턴스
- ③ 하드웨어 드라이버
- ④ 클라우드 서버를 뜻하는 용어

(4) Dockerfile의 역할은 무엇인가?

- ① 데이터베이스 테이블을 생성하는 파일
- ② 웹사이트의 HTML 템플릿 파일
- ③ 이미지를 자동으로 빌드하기 위한 명령 스크립트
- ④ 도커 로그 기록을 위한 설정

(5) 아래 중 도커 컨테이너 확인 명령은?

- ① docker logs
- ② docker ps
- ③ docker rm
- ④ docker init

(6) 도커에서 컨테이너를 백그라운드로 실행하는 옵션은?

- ① -a
- ② -e
- ③ -d
- ④ -s

(7) 컨테이너를 삭제하는 명령은?

- ① docker clear
- ② docker rm
- ③ docker stopall
- ④ docker deleteimage

(8) Docker Hub의 역할은?

- ① 서버 모니터링
- ② 이미지 저장소 제공
- ③ 웹사이트 호스팅
- ④ 네트워크 관리

(9) 도커의 장점으로 옳지 않은 것은?

- ① 어디서든 동일 환경 실행
- ② 가벼운 실행 속도
- ③ VM보다 느리고 무겁다
- ④ 배포 자동화 용이

(10) "docker run hello-world" 명령의 의미는?

- ① DB 생성
- ② 컨테이너 중지
- ③ 도커 설치 확인
- ④ 새로운 OS 설치

(11) Docker Compose의 주요 목적은?

- ① 이미지를 자동으로 압축하기 위함
- ② 여러 컨테이너를 하나의 설정 파일로 관리
- ③ 네트워크를 외부로 차단하기 위함
- ④ Apache 서버만 실행하기 위함

(12) docker-compose.yml에서 "depends_on" 역할은?

- ① 이미지 최적화
- ② 컨테이너 삭제 자동화
- ③ 실행 순서를 지정해 의존성을 관리
- ④ 저장 공간 증가

(13) Dockerfile에서 PHP 확장 pdo_mysql 설치 명령은?

- ① RUN apt install php-mysql
- ② RUN docker-install pdo
- ③ RUN php -m install pdo
- ④ RUN docker-php-ext-install pdo_mysql

(14) volumes: - ./app:/var/www/html 의 의미는?

- ① 데이터 삭제
- ② DB 백업
- ③ 호스트(app 폴더)와 컨테이너(/var/www/html)를 연결
- ④ 네트워크 차단

(15) .env 파일의 특징으로 옳은 것은?

- ① 도커에서 사용 불가
- ② 환경변수를 한 곳에서 관리
- ③ 컨테이너 실행 시 자동 삭제
- ④ PHP만 사용

(16) MySQL 초기 스키마 실행을 위한 폴더는?

- ① /var/log
- ② /usr/local/bin
- ③ /docker-entrypoint-initdb.d
- ④ /home/init-script

(17) Cloudtype 배포 시 실제 사용하는 핵심 구성은?

- ① docker-compose.yml
- ② web/Dockerfile + app/ 디렉터리
- ③ 모든 이미지 파일
- ④ host-only 네트워크

(18) PHP에서 PDO를 사용할 때 필요한 항목이 아닌 것은?

- ① DSN
- ② 사용자명
- ③ 패스워드
- ④ HTML 태그

(19) Apache 설정 변경 후 필요한 작업은?

- ① 서버 삭제
- ② `sudo systemctl restart apache2`
- ③ `apt restart php`
- ④ `docker stopall`

(20) `docker-compose up -d --build` 의 의미는?

- ① 컨테이너만 삭제
- ② 이미지 재빌드 후 백그라운드 실행
- ③ DB만 업데이트
- ④ 전체 초기화

(21) Docker Healthcheck의 목적은?

- ① 컨테이너 내부 파일 구조 변경
- ② 컨테이너가 정상 동작하는지 감시
- ③ 이미지를 삭제하기 위한 검사
- ④ 네트워크 포트를 재할당

(22) MySQL 컨테이너에서 Character-set 설정을 위해 compose에서 사용하는 옵션은?

- ① command: --set-charset utf
- ② mysql_charset utf8
- ③ command: --character-set-server=utf8mb4 --collation-server=utf8mb4_general_ci
- ④ charset=default

(23) 컨테이너가 stop 상태일 때 남아 있는 데이터는 어디 저장되는가?

- ① 컨테이너 내부 메모리
- ② /etc/init 내부
- ③ Docker Volume
- ④ RAM에만 존재

(24) 컨테이너 이름 없이 docker run 실행 시 나타나는 결과는?

- ① 오류 발생
- ② Docker가 자동으로 임의의 이름 부여
- ③ root 권한 제한
- ④ 실행 불가

(25) PHP + MySQL 게시판에서 “DB 연결 실패: getaddrinfo db failed” 오류 원인은?

- ① CSS 파일 손상
- ② DB hostname이 잘못됨(예: ‘db’ 대신 다른 값)
- ③ PHP 버전 오류
- ④ HTML 링크 누락

(26) Docker 네트워크 connect 명령의 목적은?

- ① 이미지를 다른 서버로 복제
- ② 실행 중 컨테이너를 특정 네트워크에 연결
- ③ 로그를 외부로 출력
- ④ PHP 모듈 추가

(27) Cloudtype에서 503 오류가 나는 대표 원인은?

- ① 스타일시트 경로 오류
- ② Dockerfile 내부 Apache 설정 실패 또는 컨테이너 실행 실패
- ③ HTML 주석 문제
- ④ MySQL 테이블 컬럼명 오류

(28) docker inspect 의 목적은?

- ① 컨테이너 삭제
- ② 이미지 빌드
- ③ 컨테이너 세부 정보(JSON) 확인
- ④ 웹 서버 재시작

(29) php.ini를 컨테이너 내부에 복사하는 Dockerfile 명령은?

- ① COPY php.ini /var/www/php.ini
- ② COPY php.ini /etc/apache2/
- ③ COPY php.ini /usr/local/etc/php/conf.d/99-custom.ini
- ④ COPY php.ini /home/php

(30) Docker volume rm 명령을 잘못 사용하면 발생하는 문제는?

- ① PHP 오류
- ② DB 영구 데이터 손실
- ③ HTML 문서 삭제
- ④ Apache 중지