**<php 예상문제 & 정리모음>**

**+\_+ ^\_~ ^0^ ヾ(≧▽≦\*)o ( •̀ ω •́ )✧ ヾ(•ω•`)o**

|  |
| --- |
| **<이건 무조건 시험에 나온다>**php소스 못봄 DNS = 도메인 -> ip Docker 컨테이너 접속 도커 실행 시 **-it** 옵션을 사용해서 실행 후 터미널 접속을 하도록 할 수 있다.  **ps aux** : 프로세스가 제대로 작동하고 있는지 확인하고 싶을 때 사용하는 명령어  **cd .** : 현재 디렉토리  **cd ..** : 상위 디렉토리(이동할 때 사용)  **<-d, -p, -v>**  **-d :** daemon **백그라운드**로 보낼때 사용  **-p : port (포트)**  **-v : 호스트 컴퓨터와 컨테이너 간에 공유할 디렉토리를 지정**  **-it : 터미널 입력을 위한 옵션**  **exec it : 실행중인 도커 컨테이너 접속**  **<이미지 보기(확인)>**  **docker images**  **docker image ls**  **$ : 일반 사용자**  **# : 관리자**  **$(일반사용자)에서 #(관리자)로 갈 때는 sudo su - 를 통해 갈 수 있다.**  **sudo su는 “super user do”의 줄임말.**  하나의 이미지는 여러 개의 컨테이너로 만들 수 있다.  **docker run --name ws2 httpd**  이미지 이름 앞에는 name 옵션이 온다.  **docker run**  컨테이너 만들 때  **httpd** 사용하고 싶은 이미지 이름  docker bulid -t id/imagename:tagname **.**(Dockerfile위치를 나타냄. 매우중요)  docker bulid -t progh2/nodehello:2(tag) .  docker run -d(daemon 백그라운드로 보낼때 사용) -p(port 아파트 호수와 비슷한 개념이다) poghe/nodehello:2  docker **exec** -it hellonode/bin/bash(실행중인도커컨테이너 접속)  **clear : 리눅스에서 화면 지울 때 사용**  **(cmd에서는 cls)**  [vim명령어](https://zzsza.github.io/development/2018/07/20/vim-tips/)  **dd는 줄 삭제, 라인 복사는 yy**  **해당 라인으로 이동, 라인출력 - :set nu**  **① 도커 컨테이너는 일종의 하드웨어(=소프트웨어)를 소프트웨어의 실행에 필요한 모든 것을 포함하는 완전한 파일 시스템 안에 감싼다.**  **② 도커(Docker)는 리눅스의 응용 프로그램들을 프로세스 격리 기술들을 사용해 컨테이너로 실행하고 관리하는 오픈 소스 프로젝트이다.**  **③ 도커 컨테이너는 실행 중인 환경에 관계 없이 언제나 동일하게 실행될 것을 보증한다**  **④ 도커는 리눅스에서 운영 체제 수준 가상화의 추상화 및 자동화 계층을 추가적으로 제공하지 않는다.(=제공합니다)**  **⑤ 의존적인(= 독립적인)"컨테이너"가 하나의 리눅스 인스턴스 안에서 실행할 수 있게 함으로써 가상 머신을 시작하여 유지보수해야 하는 부담을 없애준다.** |
| **<오..이건 좀 중요한데?>** **exec : 실행중인 도커 컨테이너에 접속하는 명령어**  웹브라우저 : 크롬, IE, ...  웹서버 : 파일 요청하면 도와주기  데이터베이스 : 추가하고 삭제하고  Client Side 프로그램 : Javascript. TypeScript  공개하면 안되는 거면 웹서버에서하고, 공개해도 되는것은 웹브라우저에서 한다  /home/ubuntu <- Home Drectory(~물결)  Document Root  도메인을 IP로 바꿔주는 것을 DNS(도메인 네임 서버)라고 한다   |  | | --- | | **<함쌤 마지막 족집게 강의>** | | 요약 그림판 링크  <https://classroom.google.com/c/MjkyMDQ2NzcyMDU5> |  |  | | --- | | **<기본적으로 알아두자>** | | [PHP: Hypertext Preprocessor](https://www.php.net/)  <?PHP ….. ?>  $name = “아무게” **명령어규칙**  $abc(o)  $1234(숫자 안됨)  $\_abc(언더바는 괜찮다)  $a bc(띄어쓰기안됨)  출력 명령문 : print(), echo  파일 동작확인 : phpinfo(); | |
|  |

**<php 문제정리> 😚😍😊 1일 3문제**

|  |
| --- |
| **유빈 예상문제) 난이도 3.5 (중상)** |
| **1. 다음은 '깃 저장소 생성 및 파일 커밋'을 하는 코드이다.**  **빈칸에 알맞은 문법을 작성하시오**.  git [ ]  git [ ] Dockerfile  git [ ] -m "Init Project"  init, add, push  답 : push , push ,commit => init, add, push  **2. 다음, '깃허브 저장소 원격 저장소 등록 및 푸시'를 하는 단계에서 올바르지 않은 단계를 모두 고르세요.**  1. 브랜치가 요즘에는 main 대신 master을 많이 사용하므로 main 브랜치를 master 브랜치로 변경하여 사용한다.  2. 깃 원격(remote) 저장소 등록을 한다.  3. origin 원격 저장소에 지역 저장소의 main 브렌치를 푸시한다.  답 : 3 main이 아닌 mater브렌치를 푸시한다 => +1번 mater로 완전히 바뀌었기 때문?  **\*확인문제)**  '깃허브 저장소 원격 저장소 등록 및 푸시'를 하는 단계에 빈칸에 들어갈 알맞은 문법을 쓰세요.  (단, @에 들어갈 말은 동일하다.)  git [ ] [ @ ]  git [ ] add origin [깃허브 저장소 주소]  git [ ] origin [ @ ]  답 : master => branch, remote, push @ = main  여기도 [ ] 빈 괄호 3개에 들어갈 말이랑 [@] 골뱅이 괄호 안에 들어갈 말 총 4개 쓰면 돼~  **3. 다음, '소스 파일을 깃허브로 관리'하는 단계에서 알맞지 않은 단계를 고르세요.**  1. 계정의 홈 디렉토리로 이동한다.  2. Dockerfile 디렉토리 밖에 별도로 웹 소스를 관리할 디렉토리를 하나 생성한다.  3. 기존 index.js 파일을 옮긴다.  4. 기존 Index.js 파일을 삭제한다.  5. 가상 경로가 반영하여 Docker 이미지를 다시 실행한다.  6. 동작 여부를 확인한다.  답 : 4  **4. 다음 중, 도커가 제대로 떠있는지 확인하고 싶을 때 쓰는 명령어는?**  1. pull  2. ls  3. -it  4. ps  5. cat  답 : ps  **5. 다음, 도커 컨테이너 서비스 포트를 '8080'으로, host 포트를 '80'으로 서비스를 하려고 할 때 빈칸에 알맞는 코드를 작성하세요.**  docker run -d -p [ ] progh2/hello:1  답 : 8080:80 => 80:8080  모든 문제 답 :  1번 : init, add, commit  2번 : 1, 3  2번 확인문제 : branch, remote, push / @ : main  3번 : 4  4번 : 4  5번 : 80:8080  **<2일차> 난이도 3.7 (중상)**  **자~ 저도 선우가 오늘 수업을 잘 들었는지 확인해보겠습니닷~~**  1. 다음 중, **“HTTPS”**의 기본 포트로 올바른 것을 고르세요.  1. 441  2. 442  3. 443  4. 444  5. 445  답 : 3  2. 다음 중, **“MySQL”**와 같은 말로 쓰이는 언어를 고르세요.  1. Oracle  2. MS-SQL  3. PHP  4. MariaDB  5. Linux  답 : MariaDB  3. 다음 중, **“nano”**의 기능 두가지로 올바른 것을 모두 고르세요.  1. ctrl + s  2. ctrl + p  3. ctrl + z  4. ctrl + o  5. ctrl + x  답 : 2,5 => 4,5(한개틀림)  **<3일차> 난이도 4.0 ( \*\*\*\* 별이 네 개)**  **1. 다음 중 서버 side 언어에 대한 설명으로 올바른 것을 모두 고르세요.**  1. 서버 side 언어에는 JSP, JavaScript 등이 있다.  2. 서버 side 언어에는 ASP, PHP 등이 있다.  3. 서버 side 언어에는 JavaScript, TypeScript 등이 있다.  4. 서버 side 언어는 사용자가 직접 볼 수 있다.  5. 서버 side 언어는 사용자가 직접 볼 수 없다.  답 : 2,5  **2. php 에서 글자를 출력할 때 사용한 출력문을 모두 고르세요.**  1. phpinfo()  2. print()  3. println()  4. echo  5. input()  답 : 2,4  **3. 다음 리눅스 명령어 중 저장하지 않고 나갈 때 쓰는 명령어를 고르세요.✨✨✨**  1. :q1  2. :q  3. :g!  4. q!  5. :q**!**  5  답 : 2  **<4일차>**  1. 서버에 접속하여 업데이트를 할 때 썼던 명령어로 알맞은 것을 모두 고르세요.  1. apt get update  2. apt -get update  3. apt-get upgrade  4. apt get upgrade  5. apt-get update  답 : 3,5  2. 현재 작업 디렉토리를 알고 싶을 때 쓰는 명령어로 올바른 것을 고르세요.  pwd = present working directory  1. mv  2. cp  3. aux  4. pwd  5. apm  답 : 4  3. 도커에서 이미지를 만들 때 썼던 명령어로 알맞은 것을 고르세요.  1. run  2. httpd  3. build  4. mv  5. tag  답 : 3  4. 도커 명령어를 볼 때 썼던 알맞은 명령어를 고르세요.✨✨✨  1. docker ps  2. docker  3. docker ls  4. docker ls -al  5. ls -al  답 : 4  5. 깃허브에 커밋할 때 썼던 알맞은 문법을 적으세요.✨✨✨  git [ commit ] [ **-m** ] "Init Project"  -m  첫번째 답 : apt-get update, apt-get upgrade  두번째 답 : pwd  세번째 답 : build  네번째 답 : docker  다섯번째 답: [commit] [-m]  **<5일차>**  1. 명령어 "-p"에 대한 설명(=해석)으로 알맞은 것을 모두 고르세요.✨✨  1. 포트 조정  2. 프로세스  3. public  4. push  5. publish  답 : 1,2 +5  2. 리눅스에서 화면을 지울 때 사용하는 명령어로 알맞은 것을 고르세요.  **1. clear**  2. cls  3. c  4. -c  5. -cls  답 : 2 => 1  **clear : 리눅스에서 화면 지울 때 사용**  **(cmd에서는 cls)**  3. AWS EC2 생성 및 접속 "단계 4: 스토리지 추가"에 맞지 않는 설명을 모두 고르세요.  (1~5까지 순서대로 진행됩니다.)✨✨📢  1. 하드디스크 용량 크기를 정하는 것이다.  2. 기본값은 8기가로 설정되어 있다.  3. 계정 내에서 60기가까지는 무료 용량으로 사용이 가능하다.  4. 이 중 일부 용량은 우분투 리눅스 기본 운영체제에서 사용한다.  5. 그렇기 때문에 실제 사용자가 사용 가능한 용량은 **4~6기가** 사이 정도가 된다.  답 : 3,5  1. 답 : 1, 2, 5  2. 답 : clear  3. 답 : 3 (30기가 임), 5 (4,6기가 임)  **<6일차>**  "AWS ECS 생성 및 접속하기" 단계 중 몇 단계 : [무슨내용]인지 맟추고, [ ] 괄호 안에 들어갈 말을 쓰세요.  주목할 설정 중에 [ **종료 방지 기능 활성화** ] 항목이 있는데 이것을 체크하면 이 옵션이 켜져있는 한 인스턴스를 삭제 못하도록 설정할 수 있다. 실수로 운영중인 서버를 삭제하는 것을 막도록 하는 설정으로 이 설정이 켜져있으면 인스턴스 삭제가 되지 않으며 해제해야만 인스턴스 종료가 가능하게 된다. 나중에 실제 서비스를 운영할 때 꼭 켜놓고 사용해야 할 것이다.  답 : 1. [ 인스턴스 세부설정 ], 2. [ 인스턴스 삭제방지?ㅋㅋㅋ ]  => 1. 인스턴스 세부 정보 구성 2. 종료방지기능 활성화  다음 중, PHP 언어 소개에 대한 올바른 설명으로 알맞은 것을 고르세요.  1. 주로 **동적인** 웹 페이지를 구성할 때 많이 사용된다.  2. C나 Java, JavaScript의 문법과 비슷하다. Perl  3. 전 세계적으로 모든(많은) 사람들이 PHP를 사용하고 있다.  4. 웹 프로그래밍 언어와 데이터 교환을 위해 WDOX (WDDX)표준을 지원한다.  5. 2001년 6월, 675만개 이상의 도메인을 사용한다.  답 : 5  다음 중, "orpd1 || orpd2"에 대한 설명으로 옳은 것을 고르세요.  1. 오퍼런드의 값이 FALSE이면 연산 결과가 TRUE, 오퍼런드의 값이 TRUE이면 연산 결과가 FALSE이다.  2. 두 오퍼런드의 값이 모두 TRUE일 때만 연산 결과가 TRUE, 그 외에는 연산 결과가 FALSE이다.  **3. 두 오퍼런드의 값이 모두 FALSE일 때만 연산 결과가 FALSE, 그 외에는 연산 결과가 TRUE이다.**  4. 하나의 오퍼런드는 TRUE 나머지 오퍼런드는 FALSE일 때 연산 결과가 TRUE, 그 외에는 연산 결과가 FALSE이다.  5. 하나의 오퍼런드는 FALSE 나머지 오퍼런드는 TRUE일 때 연산 결과가 FALSE, 그 외에는 연산 결과가 TRUE이다.  답 : 2 =>3  !orpd (not)  오퍼런드의 값이 FALSE이면 연산 결과가 TRUE, 오퍼런드의 값이 TRUE이면 연산 결과가 FALSE  orpd1 && orpd2 (and)  두 오퍼런드의 값이 모두 TRUE일 때만 연산 결과가 TRUE, 그 외에는 연산 결과가 FALSE  orpd1 || orpd2 (or)  두 오퍼런드의 값이 모두 FALSE일 때만 연산 결과가 FALSE, 그 외에는 연산 결과가 TRUE  orpd1 xor orpd2 (xor)  하나의 오퍼런드는 TRUE 나머지 오퍼런드는 FALSE일 때 연산 결과가 TRUE, 그 외에는 연산 결과가 FALSE |
| **선우 예상문제) 난이도 5 (최상)** |
| 1일차 **[문제1]**  **인스턴스를 만드는 순서의 세부 설명으로 옳지 않은 것은?**  1. AMI선택 : EC2 인스턴스의 베이스가 되는 운영체제를 선택하는 단계이다  2. 인스턴스 유형 선택 : 서버 인스턴스의 CPU의 개수, 메모리 크기를 정하는 단계이다  3. 인스턴스 세부 정보 구성 : 인스턴스에 대한 설정  4. 스토리지 추가 : 하드디스크 용량 크기를 정하는 것이다.(기본크기는 8기가이다)  5. 태그 추가 : 다수의 인스턴스를 관리하기위해 태그를 붙일 수 있다  6. 보안그룹 구성 : 인스턴스에 대한 방화벽을 설정하는 단계(SSH접속용 포트 제외하고 모든 포드 막힘)  7. 인스턴스 시작 검토 : 설정한 것을 확인하는 마지막 단계  답 : 4  **[문제2]**  **Node.js서비스에서 index.js소스에서 [ ]에 들어갈 것은?**  var http = require('http');  var handleRequest = function(req,res){res.end('hello'); });  var www = http. [ ](handleRequest);  ① ServerCreate  ② Create  ③ ServerMake  ④ CreateServer  ⑤ CreateRequest  답 : 4  **[문제3]**  **소스파일 깃허브로 관리하기. 순서를 적어주세요**  ⓐ디렉토리를 하나 생성한다  ⓑ가상경로를 부여한다 **?**  ⓒindex.js파일을 옮긴다  ⓓDocker 이미지를 다시 실행한다  ⓔ기존 Docker실행을 멈춘다  답 : a-c-e-b-d  ⓐ -> ⓒ -> ⓔ -> ⓑ -> ⓓ  디렉토리 하나 생성  index.js 파일 옮기기  기존 Docker 실행 멈춤  가상 경로 부여  Docker 이미지 다시 실행  **[문제4]**  **mysql에 접속해서 유저를 생성하고 권한을 부여하는 문장이다 틀린문장은?**  ①use mysql;  ②CREATE USER 'test'@'%' IDENTIFIED BY 'password';  ③ **GRANT** ALL PRIVILEGES \*.\* TO 'test'@'%';  ④FLUSH PRIVILEGES;  ⑤exit  답 : 5 -> 3  **[문제5]**  **다음 도커 명령어의 해석으로 옳지 않은것은?**  ⓐcd~ = 홈디렉토리  ⓑsuper user do = 바다야 열려라  ⓒsudo su abc - = abc계정으로 들어가  ⓓinit = 초창기(변하지 않는다)리눅스가 실행될때 처음으로 실행되는 것  ⓔESC + ctrl+z = vim 파일 **나가기**  답 : c -> e  문제 답 :  [문제1] 답 : 4 =>기본16기가가 아닌 8기가다  [문제2] 답 : ④  [문제3] 답 : ⓐ -> ⓒ -> ⓔ -> ⓑ -> ⓓ  [문제4] 답 : ③ 정답은 GRANT, REVOKE는 권한을 회수하는것  [문제5] 답 : ⓔ 파일 들어가기가 아닌 나가기다 2일차 오늘 수업을 잘 들었는지 체크해보겠습니다  **[문제1] Home Directory는 어디에 저장되어 있는가?**  ①/HOME/Document Root  ②/home/document  ③/home/ubuntu root  ④/ubuntu/ home  **⑤/home/ubuntu**  **정답 : [5]😍**  **[문제2] 다음중 사용자에게 보여지는 옳은 순서를 고르시오**  ①인터넷 -> 컴퓨터 -> PHP요청 -> 스크립트엔진 ->웹서버 ->데이터베이스->사용자  ②인터넷 -> 컴퓨터 -> PHP요청 -> 스크립트엔진 ->웹서버 ->데이터베이스->사용자  ③컴퓨터 -> 인터넷 -> 스크립트엔진 -> PHP요청 ->웹서버 ->데이터베이스->사용자  **④컴퓨터 -> 인터넷 -> PHP요청 -> 웹서버 -> 스크립트엔진 ->데이터베이스->사용자**  ⑤PHP요청 -> 컴퓨터 -> 인터넷 -> 스크립트엔진 ->웹서버 ->데이터베이스->사용자  **정답 : [4]🤗**  **[문제3] 아래 코드를 보고 틀린 부분을 고치시오**   |  | | --- | | **<?PHP**  **$low = $\_POST[‘low’];**  **$na\_me = ‘’; $price = 0;**  **$fru it = array (“사과” => 10, “배” => 20, “복숭아” => 30,**  **“수박” => 40, “감” => 50, “토마토” => 60,**  **“바나나” => 70, “키위” => 80, “파인애플” => 90,**  **“대추” => 100);**  **print “가격이” . $low . “원 이하인 과일의 목록입니다.”;**  **print “<table border=1><tr><th>이름</th><th>가격</th></tr>”;**  **foreach ($ $fru it as $na\_me => $price) {**  **if ($price <= $low)**  **print “<tr><td>” . $na\_me . “</td><td>” . $1price . “</td></tr>”;**  **}**  **print “</table>”;**  **?>** |   **정답 : [ $fru it 이다. 띄어쓰는 것은 변수명으로 올바르지 않다. ]😋** 3일차 **[문제1] PHP특징이 아닌것을 하나만 고르시오 \*\*\***  ①PHP는 **웹프로그래밍 언어**이다  ②동적인 웹페이지를 구성할때 많이 사용되고, 서버에서 동작하는 스크립트언어이다  ③PHP는 배우고 프로그래밍 하기 쉬운 언어이다.Perl과 문법이 비슷하다  **④7.0버전대와 그 전버전은 차이가 많이 난다**  ⑤**PHP는 문자의 연결자**가 ‘**.**’이다  **정답 : [4]**  1번이 답이다. php는 웹프로그래밍 언어이다.  4번이 답이 아닌 이유는 php는 7.0 버전대와 그 전 버전과 차이가 많이 나서 다운도 받기가 어렵다.  **[문제2] 다음중 도커 명령어중 맞는것을 고르시오📢**  ① docker ps : 프로세스 보여주기  ② docker run --name ws3 -p 8081:80 httpd : **8081 포트로 들어오면 80 포트 전송**  80:8081  ③ docker run httpd (이미지 이름 ) : 실행하기  ④ nano index.html : 저장하고 파일로 들어가기  ⑤ ctrl +x + y +enter : 저장하고 나가기  **정답 : [1]**  **docker ps : 실행중인 컨테이너를 보여줌**  **ps aux : 프로세스의 목록을 보여줌 ( = 프로세스 상태)**  docker run --name ws2 httpd (만들기)  docker run httpd(실행하기)  **$ : 일반 사용자**  **# : 관리자**  **$(일반사용자)에서 #(관리자)로 갈 때는 sudo su - 를 통해 갈 수 있다.**  **[문제3] 다음 사진을 보고 somefile을 실행파일에서 빼는 명령어를 쓰시오**    **정답 : [git rm --cached somefile ]** 4일차 **[문제1]**  **EC2 인스턴스 생성시, 인스턴스세부정보구성에 대해 [ ]안에 들어갈 말을 쓰시오**   |  | | --- | | **종료 방지 기능 활성화 항목이 있는데 이것을 체크하면 이 옵션이 켜져있는 한 인스턴스를 삭제 못하도록 설정할 수 있다. 실수로 운영중인 서버를 [** ① **]하는 것을 막도록 하는 설정으로 이 설정이 켜져있으면 인스턴스 삭제가 되지 않으며 [** ② **]해야만 인스턴스 종료가 가능하게 된다. 나중에 실제 서비스를 운영할 때 꼭 켜놓고 사용해야 할 것이다.** |   **정답 :** ① [ 종료 or 삭제 ] ② [ 해제 ]  **[문제2] 다음 순서를 읽고, [ ]에 들어갈 알맞은 명령어를 고르시오.**   |  | | --- | | 1. **계정의 홈 디렉토리로 이동한다.** 2. **Dockerfile 디렉토리 밖에 별도로 웹 소스를 관리할 디렉토리를 하나 생성한다.** 3. **기존 index.js 파일을 옯긴다.** 4. **기존 Docker 실행을 멈춘다.** 5. **가상 경로가 반영하여 Docker 이미지를 다시 실행한다.** 6. **동작 여부를 확인한다.**   cd ~  [ ① ] www  [ ② ] hellonode/index.js www/  docker [ ③ ]-v/root/www:/www progh2/hellonode:3 |   **정답 :** ① [ mkdir ] ② [ mv ] ③ [ run ]  **[문제3] 도커 명령어중 맞는 것을 모두 고르시오**  ① docker ps - a: 컨테이너 리스트 (컨테이너 상태 리스트)  ② docker images : docker에 존재하는 이미지 목록 보기  ③ docker rm c1d25ba5c260 : CONTAINER TAG를 이용하여 삭제 (컨테이너 ID를 이용해서 삭제)  ④ docker rm -f / docker ps -a -q : 도커 컨테이너를 모두 삭제해준다  ⑤ apt-get install -y php5.6 : php5.6으로 업데이트 한다 ( 설치해준다.)  **정답** : [ 2 ] , [ 5 ]  4 5일차 **[문제1] Dockerfile을 수정할 때 알맞을 경로를 고르시오 \*\*\*\*\*\*\*\*\***    ① ubuntu/home/example  ② ubuntu/example/home  ③ home/ubuntu/example  ④ ununtu/~/example  ⑤ ~/example/ubuntu  **정답** : [ 3 ]  **[문제2] 다음 도커와 관련된 설명을 읽고 옳은 것을 모두 고르시오. \*\*\*\*\*\*\*\*\*\*\***  **① 도커 컨테이너는 일종의 하드웨어(=소프트웨어)를 소프트웨어의 실행에 필요한 모든 것을 포함하는 완전한 파일 시스템 안에 감싼다.**  **② 도커(Docker)는 리눅스의 응용 프로그램들을 프로세스 격리 기술들을 사용해 컨테이너로 실행하고 관리하는 오픈 소스 프로젝트이다.**  **③ 도커 컨테이너는 실행 중인 환경에 관계 없이 언제나 동일하게 실행될 것을 보증한다**  **④ 도커는 리눅스에서 운영 체제 수준 가상화의 추상화 및 자동화 계층을 추가적으로 제공하지 않는다.(=제공합니다)**  **⑤ 의존적인(= 독립적인)"컨테이너"가 하나의 리눅스 인스턴스 안에서 실행할 수 있게 함으로써 가상 머신을 시작하여 유지보수해야 하는 부담을 없애준다.**  **정답** : [ 2,3 ]  내일은 나무위키 참조한다 하하핳  [vim명령어](https://zzsza.github.io/development/2018/07/20/vim-tips/)  **dd는 줄 삭제, 라인 복사는 yy**  **해당 라인으로 이동, 라인출력 - :set nu**  **[문제3] OX 퀴즈**  **⚪ vim창에서 입력을 하기 위해 써야하는 것은 i 이다 [o]**  **⚪ ESC를 누르면 명령 모드가 된다[o]**  **⚪ dd는 한줄을 복사해 밑에 줄에 넣는 명령어이다[x] = dd는 줄 삭제, 라인 복사는 yy,**  **⚪ :{line number}은 라인을 출력하는 명령어이다[o] x = 해당 라인으로 이동, 라인출력 - :set nu**  **⚪ ^와$는 라인의 처음, 끝으로 이동하는 명령어이다[x] o** 6일차 **[문제1] AWS EC2에 php설치 순서에 알맞지 알맞지 않은것을 고르고 알맞은 순서로 바꾸시오. (두개)**   |  | | --- | | 1️⃣ Dockerfile수정 -> 2️⃣ 도커컨테이너확인, 도커 컨테이너 모두 삭제-> 3️⃣ software-properties-commn, php5.6 repository, php5.6설치-> 4️⃣ 다시 빌드 -> 5️⃣ 지역입력못함(docker image파일을 빌드중이기때문 -> 6️⃣docker파일 수정->7️⃣Ctrl+z를 이용해 종료->8️⃣정상적으로 빌드 성공 |   **정답** : [ ] → [ ]  1<->2 / 6<->7  1,7  **[문제2] php설치관련 o/x퀴즈**  **⚪ docker images<none>, 사용하지 않은 이미지파일을 칭하는말 이다.[o]**  **⚪ docker rmi -f{image id}를 이용해서 이미지파일을 삭제할 수 있다[x]**  **⚪ 이미지 파일을 컨테이너에 담아서 실행하는 명령어는 이것이다**  **docker run -p 80:80 -v /home/ubuntu/example/html:/var/www/html example[o] =>[x]**  **⚪ 마운트를 진행해주면 html파일이 추가 될때마다 /var/www/html파일을 바뀐다[o]**  **/var/www/docker.html**  **-v 마운트를 진행한다. 의 약자**  **마운트를 진행해 주면 호스트에 html폴더에 파일을 추가하면 컨테이너의 /var/www/html 폴더에 파일을 추가한 것과 같은 효과가 된다.**  **[문제3] 다음 사진을 보고 물음에 답하시오**     1. **-y 가 뜻하는 것은 무엇인가?** 2. **CMD 코드가 뜻하는 것은 무엇인가?** 3. **80번포트를 사용하는 이유는 무엇인가?**   **정답** : (1) [ yes ]  (2) [ -d는 데몬, 백그라운드 상태로 바꿔준다. ]  컨테이너는 특정 작업을 완료하고 나서 곧바로 종료되기 때문에 기본적으로 apachectl가 항상 구동되는 상태로 만들어 주기 위해 설정한다(데몬 상태로 만들어 준다).  (3) [ 기본 포트이기 때문. ] apache의 기본 포트는 80포트이기 때문이다.  **apachectl** 은 아파치 하이퍼텍스트 전송 프로토콜 (HTTP) 서버의 앞단이다.  => 이 프로그램은 관리자가 아파치 httpd 데몬을 조정하도록 돕는다. |