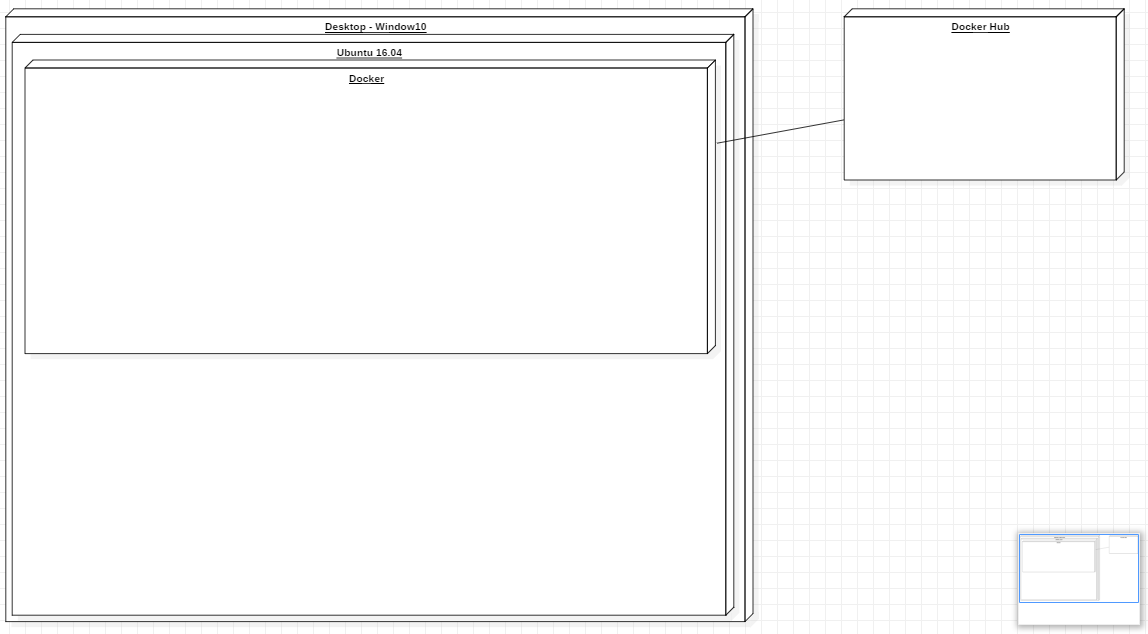
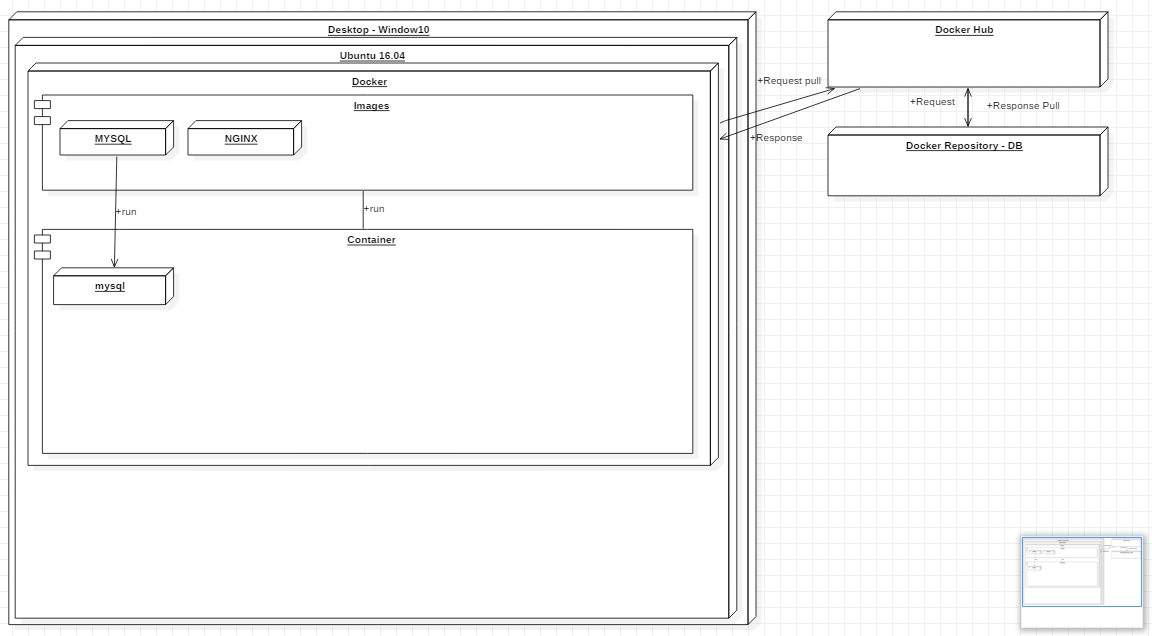
1. 내컴>vmware>ubuntu18에 docker 설치 (참조문서 : <https://futurecreator.github.io/2018/11/16/docker-container-basics/>)

|  |
| --- |
| 내컴>vmware>ubuntu18> curl –fsSL <https://get.docker.com/> | sudo sh |



1. 도커에 mysql 5.7 설치, 가동

|  |
| --- |
| 내컴>vmware>ubuntu18> docker run -d -p 3306:3306  -e MYSQL\_ALLOW\_EMPTY\_PASSWORD=true  --name mysql mysql:5.7 |



1. mysql이 돌고 있는 컨테이너(ex Debian9)에 들어가서 bash를 실행하기

|  |
| --- |
| 내컴>vmware>ubuntu18> docker exec -it mysql /bin/bash |

1. mysql 클라이언트를 실행하기

|  |
| --- |
| 내컴>vmware>ubuntu18>docker>Debian9> mysql -uroot |

1. mysql 명령해보기

|  |
| --- |
| 내컴>vmware>ubuntu18>docker>Debian9>mysql> show databases; |

1. mysql client 빠져 나오기

|  |
| --- |
| 내컴>vmware>ubuntu18>docker>Debian9>mysql> quit; |

1. mysql이 돌고 있는 컨테이너(ex Debian9) 빠져 나오기

|  |
| --- |
| 내컴>vmware>ubuntu18>docker>Debian9> exit |

1. 현재 머쉰 확인하기

|  |
| --- |
| 내컴>vmware>ubuntu18> cat /etc/issue |

1. 도커에 현재 동작하고 있는 컨테이너 확인해 보기

|  |
| --- |
| 내컴>vmware>ubuntu18> docker ps |

1. 도커 mysql client로 바로 들어가보기

|  |
| --- |
| 내컴>vmware>ubuntu18> docker exec –it mysql mysql -uroot |

1. mysql에 새 db와 테이블을 만들어보기

|  |
| --- |
| 내컴>vmware>ubuntu18>docker>Debian9>mysql> create database test;  내컴>vmware>ubuntu18>docker>Debian9>mysql> use test;  내컴>vmware>ubuntu18>docker>Debian9>mysql> create table member( no int primary key auto\_increment, name varchar(30), id varchar(10), pw varchar(10));  내컴>vmware>ubuntu18>docker>Debian9>mysql> insert into member(name,id,pw) values('jes','secure','block');  내컴>vmware>ubuntu18>docker>Debian9>mysql> select \* from member;  내컴>vmware>ubuntu18>docker>Debian9>mysql> quit;  내컴>vmware>ubuntu18> |

1. mysql 컨테이너를 종료했다 다시 올려보기

|  |
| --- |
| 내컴>vmware>ubuntu18> docker ps  내컴>vmware>ubuntu18> docker stop mysql  내컴>vmware>ubuntu18> docker ps  내컴>vmware>ubuntu18> docker start mysql  내컴>vmware>ubuntu18> docker ps  내컴>vmware>ubuntu18> docker exec –it mysql mysql –uroot  내컴>vmware>ubuntu18>docker>Debian9>mysql> show databases;  (이전에 했던 작업이 저장되어 있는 것을 확인한다)  내컴>vmware>ubuntu18>docker>Debian9>mysql> quit;  내컴>vmware>ubuntu18> |

1. mysql 컨테이너 이미지를 삭제했다 다시 올려보기

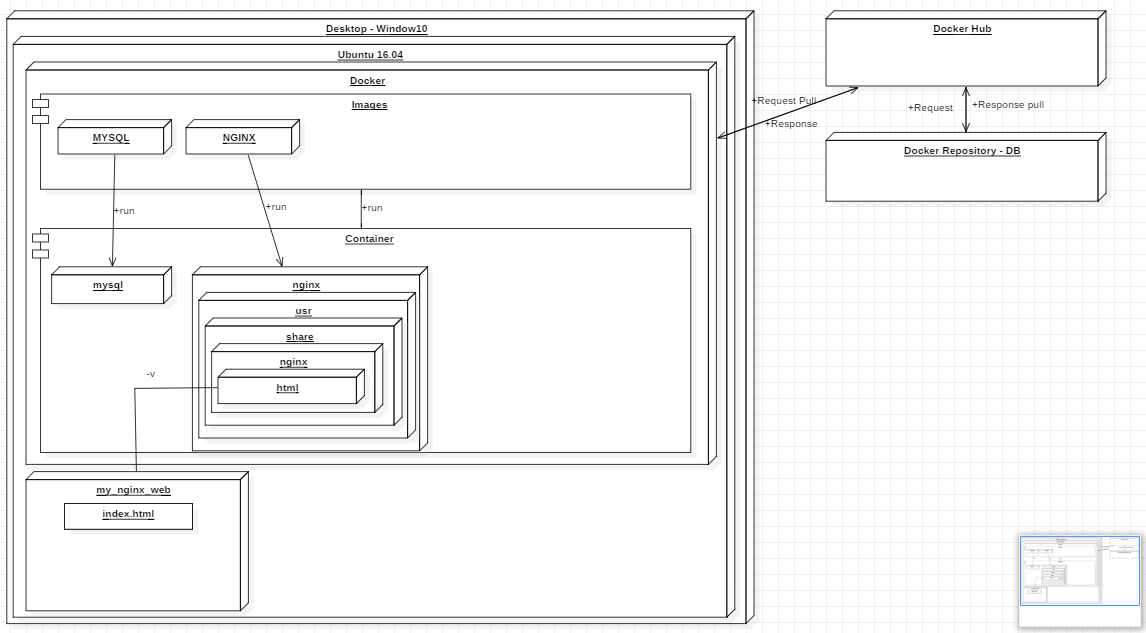
|  |
| --- |
| 내컴>vmware>ubuntu18> docker ps  내컴>vmware>ubuntu18> docker stop mysql  내컴>vmware>ubuntu18> docker ps  내컴>vmware>ubuntu18> docker images  내컴>vmware>ubuntu18> docker rmi –f <mysql id>  내컴>vmware>ubuntu18> docker images  내컴>vmware>ubuntu18> docker run -d -p 3306:3306  -e MYSQL\_ALLOW\_EMPTY\_PASSWORD=true --name mysql mysql:5.7  내컴>vmware>ubuntu18> docker exec –it mysql mysql –uroot  내컴>vmware>ubuntu18>docker>Debian9>mysql> show databases;  (다시 받았으므로 test DB와 member 테이블은 없다) |

1. mysql에 새로운 db를 추가하고 이 이미지를 커밋하기

|  |
| --- |
| 내컴>vmware>ubuntu18>docker>Debian9>mysql> create database test;  내컴>vmware>ubuntu18>docker>Debian9>mysql> quit;  내컴>vmware>ubuntu18> docker stop mysql  내컴>vmware>ubuntu18> docker ps  내컴>vmware>ubuntu18> docker images  내컴>vmware>ubuntu18> docker commit mysql mysql:test\_db  내컴>vmware>ubuntu18> docker images  내컴>vmware>ubuntu18> docker run -d -p 3306:3306  -e MYSQL\_ALLOW\_EMPTY\_PASSWORD=true --name mysql2 mysql:test\_db  내컴>vmware>ubuntu18> docker exec –it mysql2 mysql –uroot  내컴>vmware>ubuntu18>docker>Debian9>mysql> show databases;  **(변경 사항이 저장되어 있지 않다!!!!!!!==>컨테이너의 볼륨을 빼고 커밋되는데, 호스트의 볼륨을 사용하는 것이 아니라 컨테이너의 볼륨을 사용하는 거라서)** |

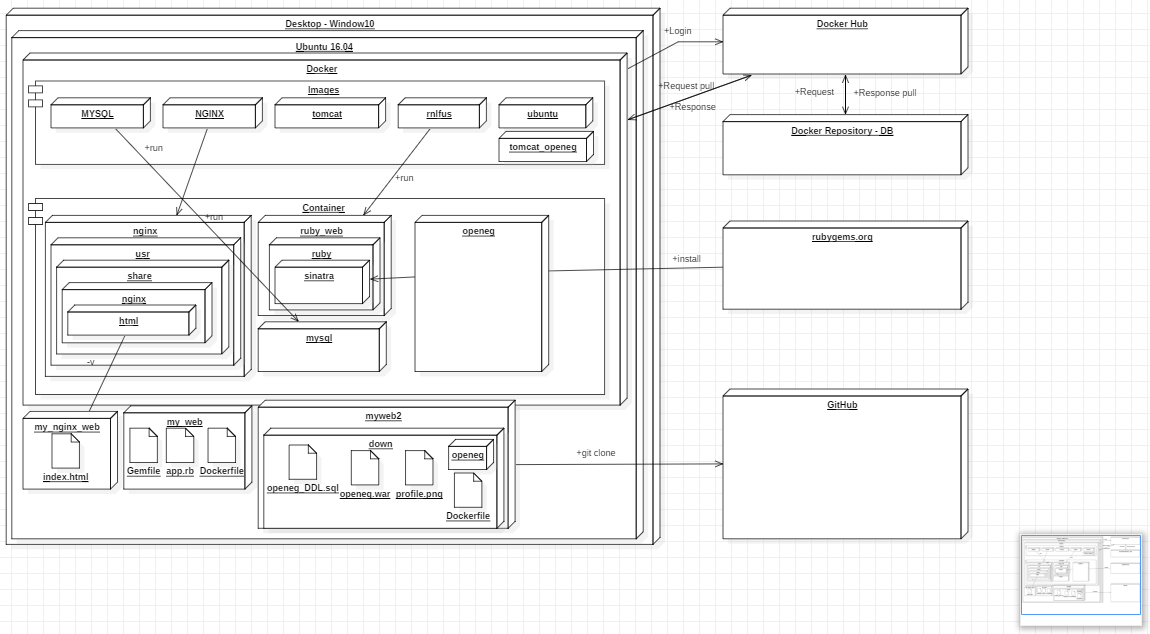
1. 도커에 nginx 설치하기

|  |  |
| --- | --- |
| 내컴>vmware>ubuntu18> mkdir my\_nginx\_web  내컴>vmware>ubuntu18> cd my\_nginx\_web  내컴>vmware>ubuntu18> docker pull nginx  내컴>vmware>ubuntu18> docker run --name nginx -v $(pwd):/usr/share/nginx/html:ro -d -p 80:80 -p 443:443 nginx  내컴>vmware>ubuntu18> vi index.html   |  | | --- | | <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  </head>  <body>  hi~ 은수 ^^  </body>  </html> |   내컴>vmware>ubuntu18> 웹브라우저로 localhost 로 확인한다 |



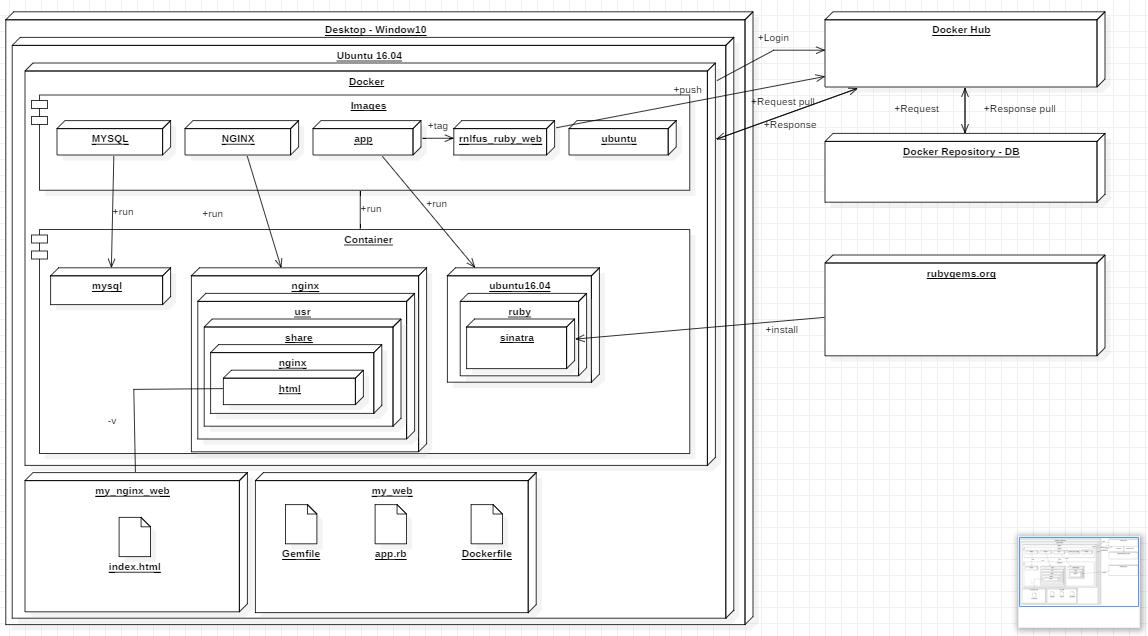
1. 루비로 간단한 웹어플리케이션 만들어보기

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 내컴>vmware>ubuntu18> mkdir myweb  내컴>vmware>ubuntu18> cd myweb  내컴>vmware>ubuntu18> vi Gemfile  source 'https://rubygems.org'  gem 'sinatra'  내컴>vmware>ubuntu18> vi app.rb  require 'sinatra'  require 'socket'  get '/' **do**  “hello jes”  **end**  내컴>vmware>ubuntu18> vi Dockerfile   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | |  | |  | # 1. 우분투 설치  FROM ubuntu:16.04 | |  | MAINTAINER jes@jes.com | |  | RUN apt-get -y update | |  |  | |  | # 2. ruby 설치 | |  | RUN apt-get -y install ruby | |  | RUN gem install bundler | |  |  | |  | # 3. 소스 복사 | |  | COPY . /usr/src/app | |  |  | |  | # 4. Gem 패키지 설치 (실행 디렉토리 설정) | |  | WORKDIR /usr/src/app | |  | RUN bundle install | |  |  | |  | # 5. Sinatra 서버 실행 (Listen 포트 정의) | |  | EXPOSE 4567 | |  | CMD bundle exec ruby app.rb -o 0.0.0.0 | |   내컴>vmware>ubuntu18> docker build –t app . (t는 tag라는 뜻, . 잊지말기)  내컴>vmware>ubuntu18> docker images (우분투 16.04와 app이 보임)  내컴>vmware>ubuntu18> docker run –d –p 8888:4567 app  내컴>vmware>ubuntu18> 웹브라우저로 <http://localhost:8888> 로 확인 |



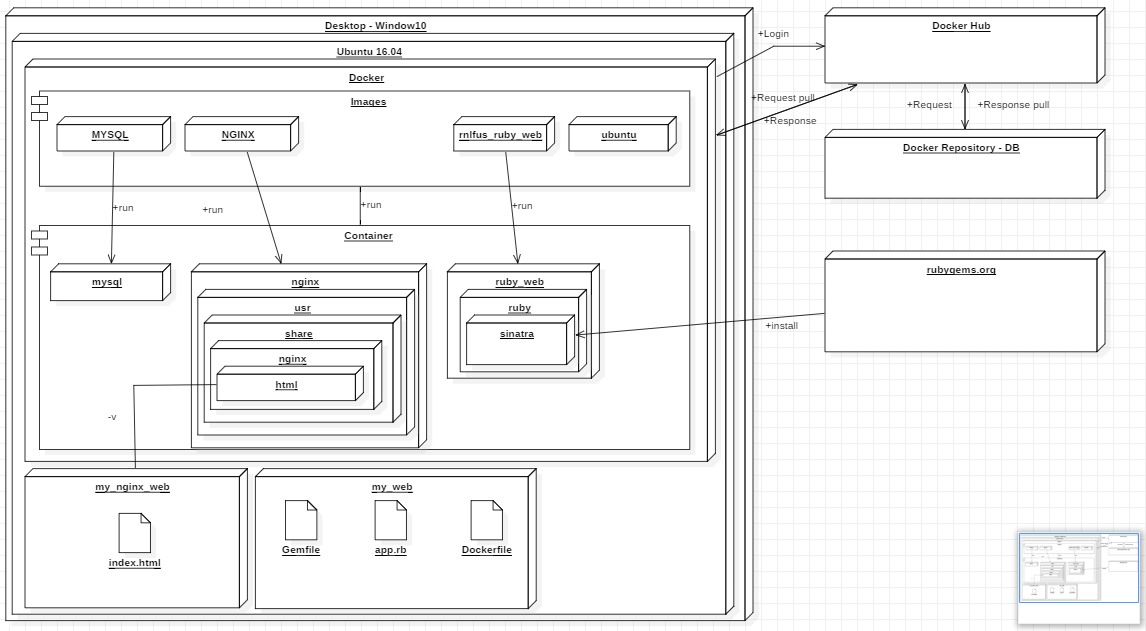
1. 도커 레지스트리에 루비 웹 배포(push)하기

|  |
| --- |
| <https://hub.docker.com/> 사이트로 가서 sign up 하기  내컴>vmware>ubuntu18> docker login  내컴>vmware>ubuntu18> docker tag app javanism/ruby\_web:1  내컴>vmware>ubuntu18> docker push javanism/ruby\_web:1 |



1. 도커 레지스트리에서 내 루비웹 이미지 받아 실행해보기

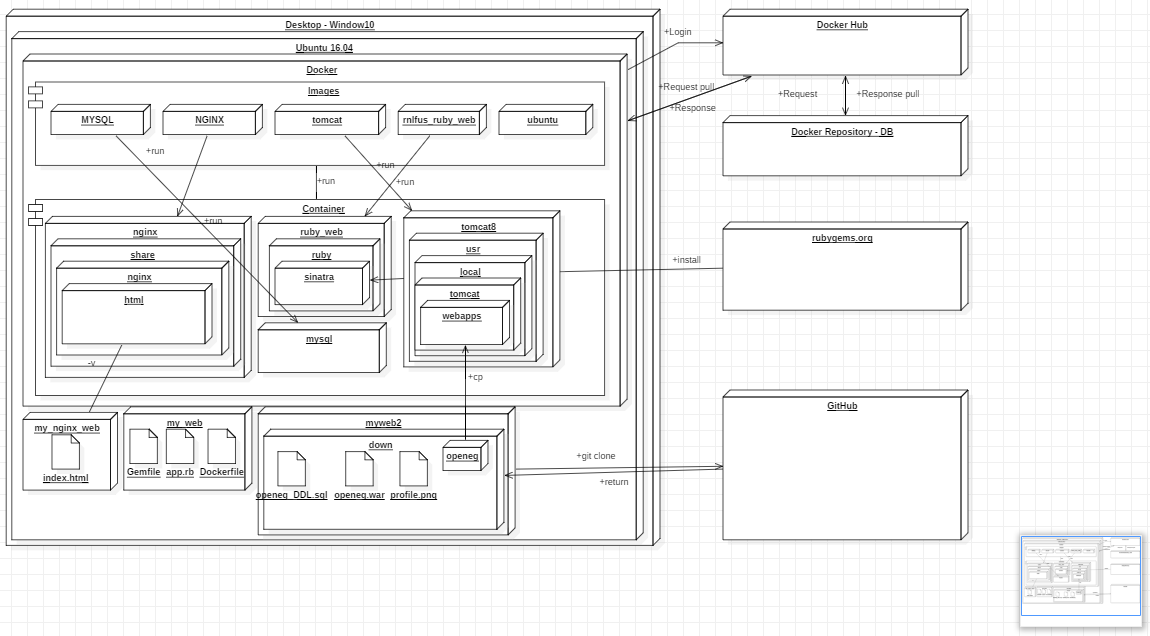
|  |
| --- |
| 내컴>vmware>ubuntu18> docker stop app  내컴>vmware>ubuntu18> docker rmi –f app  내컴>vmware>ubuntu18> docker images  내컴>vmware>ubuntu18> docker pull javanism/ruby\_web:1  내컴>vmware>ubuntu18> docker images  내컴>vmware>ubuntu18> docker run –d –p 9999:4567 --name ruby\_web javanism/ruby\_web:1  내컴>vmware>ubuntu18> 웹브라우저로 http://localhost:9999 로 확인 |



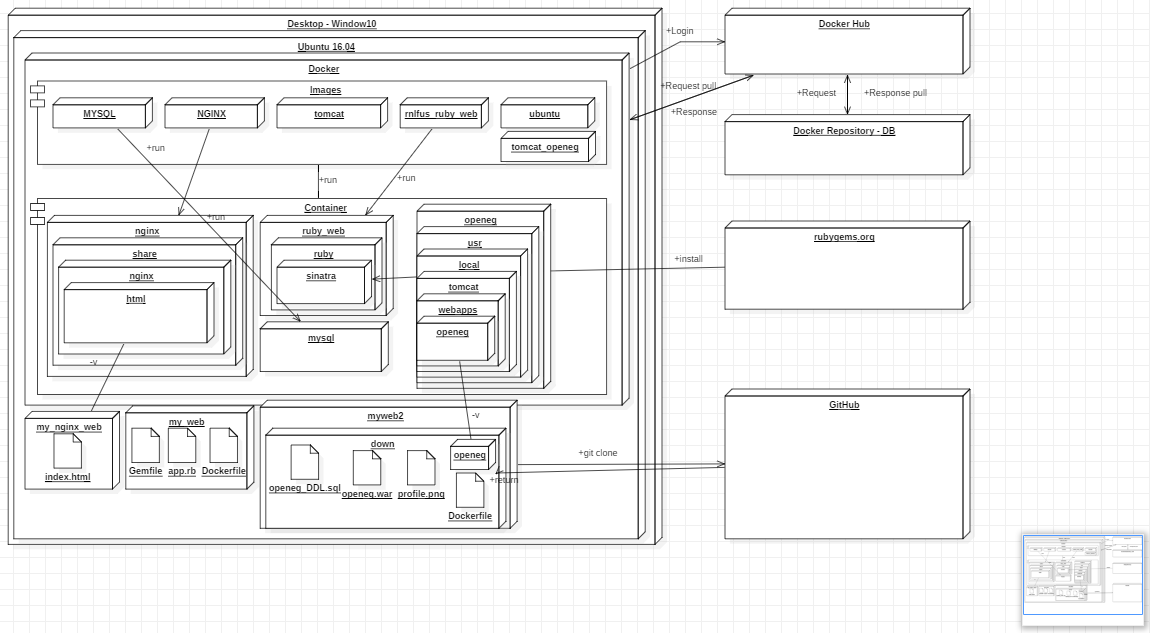
1. 도커 레지스트리에서 tomcat 8 이미지 받아 웹어플리케이션 추가해보기

|  |  |
| --- | --- |
| 내컴>vmware>ubuntu18> docker pull tomcat:8  내컴>vmware>ubuntu18> docker images  내컴>vmware>ubuntu18> docker run –d –-name tomcat8 –p 8080:8080 tomcat:8  내컴>vmware>ubuntu18> docker ps  내컴>vmware>ubuntu18> 웹브라우저로 <http://localhost:8080> 로 확인  내컴>vmware>ubuntu18> mkdir myweb2  내컴>vmware>ubuntu18> cd myweb2  내컴>vmware>ubuntu18> git clone https://github.com/javanism/down  내컴>vmware>ubuntu18> cd down  내컴>vmware>ubuntu18> mkdir openeg | unzip openeg.war –d openeg/  내컴>vmware>ubuntu18> docker cp openeg tomcat8:/usr/local/tomcat/webapps/  내컴>vmware>ubuntu18> docker exec –it tomcat8 /bin/bash  내컴>vmware>ubuntu18>docker>Devian9> cd webapps  내컴>vmware>ubuntu18>docker>Devian9> ls (openeg가 복사되었는지 확인)  내컴>vmware>ubuntu18> 웹브라우저로 <http://localhost:8080>/openeg 로 확인 (index 화면은 잘보임)  내컴>vmware>ubuntu18>docker>Devian9> Ctrl+P Ctrl+Q (컨테이너 detach)  내컴>vmware>ubuntu18> vi header.jsp (33행을 고쳐본다)  내컴>vmware>ubuntu18> 이것이 웹브라우저에 반영되지 않는다 ( 또한 컨테이너 안에서 변경 작업은 더 불편)  내컴>vmware>ubuntu18> docker rm –f tomcat8  내컴>vmware>ubuntu18> vi Dockerfile (down 폴더 아래서)   |  | | --- | | FROM tomcat:8  MAINTAINER jes@jes.com  COPY openeg /usr/local/tomcat/webapps/openeg |   내컴>vmware>ubuntu18> docker build –t tomcat\_openeg . (t는 tag라는 뜻, . 잊지말기)  내컴>vmware>ubuntu18> docker images (tomcat\_openeg가 보임)  내컴>vmware>ubuntu18> docker run –d –-name openeg –p 8080:8080 –volume=$(pwd)/openeg:/usr/local/tomcat/webapps/openeg tomcat\_openeg  내컴>vmware>ubuntu18> docker exec –it openeg /bin/bash  내컴>vmware>ubuntu18>docker>Devian9> cd webapps  내컴>vmware>ubuntu18>docker>Devian9> ls (openeg가 복사되었는지 확인)  내컴>vmware>ubuntu18> 웹브라우저로 <http://localhost:8080>/openeg 로 확인 (index 화면은 여전히 잘보임)  내컴>vmware>ubuntu18> cd openeg/WEB-INF/board/ (현재 down 폴더에서)  내컴>vmware>ubuntu18> vi header.jsp (33행을 고쳐본다)  내컴>vmware>ubuntu18> 웹브라우저에 잘 반영된다  내컴>vmware>ubuntu18> docker rm –f openeg  내컴>vmware>ubuntu18> docker ps |

내컴>vmware>ubuntu18> docker rm –f tomcat8 이전

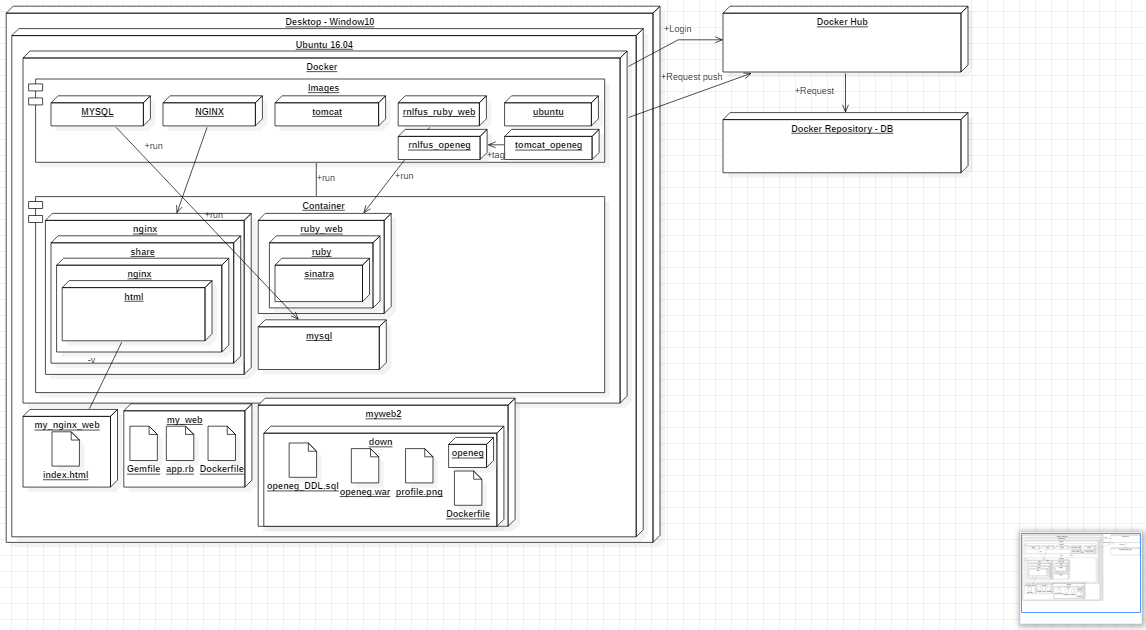


내컴>vmware>ubuntu18> docker rm –f tomcat8 이후



1. 도커 레지스트리에 tomcat\_openeg 컨테이너 배포(push)하기

|  |
| --- |
| <https://hub.docker.com/> 사이트로 가서 sign up 하기  내컴>vmware>ubuntu18> docker login  내컴>vmware>ubuntu18> docker tag tomcat\_openeg javanism/openeg:1  내컴>vmware>ubuntu18> docker push javanism/openeg:1 |



1. 도커 레지스트리에서 내 openeg 웹 이미지 받아 실행해보기

|  |
| --- |
| 내컴>vmware>ubuntu18> docker stop (tomcat\_openeg 관련 이미지)  내컴>vmware>ubuntu18> docker rmi –f (tomcat포함 tomcat\_openeg 관련 이미지)  내컴>vmware>ubuntu18> docker images  내컴>vmware>ubuntu18> docker pull javanism/openeg:1 (tomcat까지 받아야 하므로 오래 걸림)  내컴>vmware>ubuntu18> docker images (tomcat이 따로 보이진 않음)  내컴>vmware>ubuntu18> docker run –d –-name openeg –p 8080:8080 –volume=$(pwd)/openeg:/usr/local/tomcat/webapps/openeg javanism/openeg:1  내컴>vmware>ubuntu18> 웹브라우저로 <http://localhost:8080>/openeg 로 확인  내컴>vmware>ubuntu18> cd openeg/WEB-INF/board/ (현재 down 폴더에서)  내컴>vmware>ubuntu18> vi header.jsp (33행을 고쳐본다)  내컴>vmware>ubuntu18> 웹브라우저에 잘 반영된다  내컴>vmware>ubuntu18> 웹브라우저에 admin/openeg로 로그인 해보면 500에러가 발생한다. 이것은 mysql과 연동되고 있지 않아서이다 |

