RichMinime

포팅 매뉴얼



조은비 김경희 문홍웅

이주희 배찬일 조윤영

1. 사용한 JVM, 웹서버, WAS 제품 등의 종류와 설정 값, 버전(IDE포함) 기재

OS : window 10

IDE : Intellij, VSCode

Build Automation Tool : Gradle(8.1.1)

Frontend : Node.js(v18.17.0), React, Flutter

Backend : Spring boot, spring data jpa, spring security

Database : h2(develop), Mariadb(deploy)

JVM version : azul-11

WS: nginx(reverse proxy)

WAS : tomcat(spring-boot)

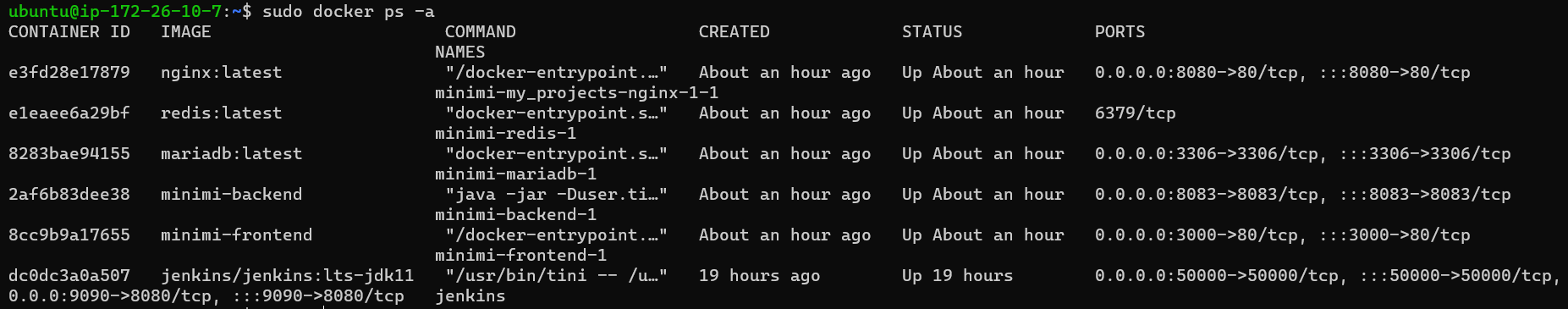
CI/CD : Jenkins

1. 빌드 시 사용되는 환경 변수 등의 주요 내용 상세 기재
2. VERSION=$(curl—silent https://api.github.com/repos/docker/compose/releases/latest | jq .name -r) : Docker-compose의 최신 릴리즈 정보
3. DESTINATION=/usr/bin/docker-compose : docker-compose 실행 파일이 저장될 위치
4. Docker-compse –p frontend build : Docker-compose는 보통 docker-compose.yml이 저장 된 디렉토리 명을 환경변수로 설정하는데 본 프로젝트에서는 docker-compose.yml이 root에 있어 이러한 자동 환경변수 설정이 되지 않기 때문에 –p 를 이용해 frontend라고 수동으로 명명해주었다

배포 순서

1. Sudo docker-compose –p frontend build
2. Sudo docker-compose –p frontend up –d
3. Sudo service nginx restart

현재 띄우고 있는 도커 컨테이너 목록



1. 배포 시 특이사항 기재
2. Docker 사용 : Backend, Frontend, Nginx, Jenkins, YoloV5, OpenVidu, DB를 각각의 Docker Container로 관리하여 독립성을 보장시켰다.
3. Docker-compose 사용 : Docker-compose를 사용하여 각각의 Dockerfile을 통해 여러 개의 컨테이너를 build 후 생성하여 다중 컨테이너 관리, 설정의 일관성, 개발 효율성을 증대시켰다.
4. Jenkins pipeline 사용 : jenkins pipeline으로 한 프로젝트에 있는 backend단과 frontend단을 동시에 자동 배포 하여 배포 환경 테스트 효율성을 증가시켰다.
5. Nginx의 reverse froxy 사용 : 가비아에서 구입한 doggy-yummy 도메인 뒤에 /, /api, /v1 경로로 reverse froxy 하여 각각 front, back, object detection 단으로 보내주었다.
6. DB 접속 정보 등 프로젝트(ERD)에 활용되는 주요 계정 및 프로퍼티가 정의된 파일 목록
7. application.yml

spring:  
 profiles:  
*# active: back-develop* active: develop

1. application-back-develop.yml (로컬용)

management:  
 health:  
 diskspace:  
 enabled: true  
 default:  
 enabled: true  
 db:  
 enabled: true  
 servlet:  
 context-path: /manage  
spa:  
 default-file: /dist/index.html5  
spring:  
 web:  
 resources:  
 static-locations: classpath:/dist/  
 add-mappings: "false"  
 jpa:  
 properties:  
 hibernate:  
 format\_sql: true  
 show\_sql: true  
 dialect: org.hibernate.dialect.H2Dialect  
 hibernate:  
 ddl-auto: create  
 defer-datasource-initialization: true  
 sql:  
 init:  
 mode: *always* datasource:  
 driver-class-name: org.h2.Driver  
 password: ''  
 username: sa  
 hikari:  
 jdbc-url: jdbc:h2:tcp://localhost/~/richMinime  
 h2:  
 console:  
 path: /h2-console  
 enabled: true  
 data:  
 web:  
 pageable:  
 one-indexed-parameters: true  
 mvc:  
 throw-exception-if-no-handler-found: true  
 devtools:  
 livereload:  
 enabled: true  
 mail:  
 host: smtp.gmail.com  
 port: 587  
 username: richminime@gmail.com  
 password: 'tgtzmsgpaabqxhkh'  
 properties:  
 mail.smtp.auth: true  
 mail.smtp.starttls.enable: true  
 redis:  
 host: localhost  
 port: 6379  
logging:  
 level:  
 org:  
 hibernate.SQL: DEBUG  
 hibernate.type.descriptor.sql.BasicBinder: TRACE  
 spring.boot: DEBUG  
 spring.security: DEBUG  
 spring.web: DEBUG  
 apache.tiles: INFO  
 root: INFO  
 com.samsung.security: DEBUG  
server:  
 error:  
 include-stacktrace: *never* servlet:  
 encoding:  
 enabled: true  
 force: true  
 charset: UTF-8  
 contextPath: /api  
 address: 0.0.0.0  
 port: 8080  
 compression:  
 enabled: true  
 mime-types: application/json,application/xml,text/html,text/xml,text/plain,application/javascript,text/css  
build:  
 date: "@build.date@"  
jwt:  
 expiration: '1296000000'  
 cookieName: refresh-token  
 secret: dyAeHubOOc8KaOfYB6XEQoEj1QzRlVgtjNL8PYs1A1tymZvvqkcEU7L1imkKHeDa  
springfox:  
 documentation:  
 swagger:  
 use-model-v3: false  
rsa:  
 private-key:   
 public-key: MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCaLpuN9VSIG2GZDGO6iGnD6dEh8eLijmK+EthBVXN8zI4B7o2fYIffxpyLMwieJzgHmw9IM347qfoEbP+L1FhTDM3OHdUxHTHRda2tjim+WLQmKkffHLt18MxaUtDjb16OrTJt6xhKQK3QRhC1owX5rSpLIc8ROqLai9OFlSbNQQIDAQAB  
codef:  
 auth-header: 'Basic MjFjZTRiYTgtNTUzNy00NTRmLTg3YmUtN2RhYWQyOGVjYjFmOjA0MTZmODdjLWI4YTYtNGJjMS05ZDM2LTJlYTgwOTVlMTNjNA=='  
gpt:  
 api-key: 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

1. application-develop.yml(배포용)

management:  
 health:  
 diskspace:  
 enabled: true  
 default:  
 enabled: true  
 db:  
 enabled: true  
 servlet:  
 context-path: /manage  
spa:  
 default-file: /dist/index.html5  
spring:  
 web:  
 resources:  
 static-locations: classpath:/dist/  
 add-mappings: false  
 jpa:  
 properties:  
 hibernate:  
 format\_sql: true  
 show\_sql: true  
 hibernate:  
 ddl-auto: none  
 defer-datasource-initialization: true  
 sql:  
 init:  
 mode: *never* datasource:  
 url: jdbc:mariadb://mariadb:3306/richminime  
 username: root  
 password: richminime  
 driver-class-name: org.mariadb.jdbc.Driver  
 data:  
 web:  
 pageable:  
 one-indexed-parameters: true  
 mvc:  
 throw-exception-if-no-handler-found: true  
 devtools:  
 livereload:  
 enabled: "true"  
 mail:  
 host: smtp.gmail.com  
 port: 587  
 username: richminime@gmail.com  
 password: 'tgtzmsgpaabqxhkh'  
 properties:  
 mail.smtp.auth: true  
 mail.smtp.starttls.enable: true  
 redis:  
 host: redis  
 port: 6379  
logging:  
 level:  
 org:  
 hibernate.SQL: DEBUG  
 hibernate.type.descriptor.sql.BasicBinder: TRACE  
 spring.boot: DEBUG  
 spring.security: DEBUG  
 spring.web: DEBUG  
 apache.tiles: INFO  
 root: INFO  
 com.samsung.security: DEBUG  
server:  
 error:  
 include-stacktrace: *never* servlet:  
 encoding:  
 enabled: true  
 force: true  
 charset: UTF-8  
 contextPath: /api  
 address: 0.0.0.0  
 port: 8083  
 compression:  
 enabled: true  
 mime-types: application/json,application/xml,text/html,text/xml,text/plain,application/javascript,text/css  
build:  
 date: "@build.date@"  
jwt:  
 expiration: '1296000000'  
 cookieName: refresh-token  
 secret: dyAeHubOOc8KaOfYB6XEQoEj1QzRlVgtjNL8PYs1A1tymZvvqkcEU7L1imkKHeDa  
springfox:  
 documentation:  
 swagger:  
 use-model-v3: false  
rsa:  
 private-key:   
 public-key: MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCaLpuN9VSIG2GZDGO6iGnD6dEh8eLijmK+EthBVXN8zI4B7o2fYIffxpyLMwieJzgHmw9IM347qfoEbP+L1FhTDM3OHdUxHTHRda2tjim+WLQmKkffHLt18MxaUtDjb16OrTJt6xhKQK3QRhC1owX5rSpLIc8ROqLai9OFlSbNQQIDAQAB  
codef:  
 auth-header: 'Basic MjFjZTRiYTgtNTUzNy00NTRmLTg3YmUtN2RhYWQyOGVjYjFmOjA0MTZmODdjLWI4YTYtNGJjMS05ZDM2LTJlYTgwOTVlMTNjNA=='  
gpt:  
 api-key: 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