Luck Quiz?

목 차

@ 2023. A707 갯벌소프트
all right reserved.

I. 빌드 및 배포

1. 개발 환경

Server1: AWS EC2 Ubuntu 20.04 LTS

Server2: AWS EC2 Ubuntu 20.04 LTS
(EC2[xlarge] - CPU : 4vCPUs, RAM : 16GB, SSD : 320GB SSD, HDD : 6 TB)

Server3: AWS EC2 Ubuntu 20.04 LTS
(EC2[t2.2xlarge] - CPU : 8vCPUs, RAM : 32GB, SSD : 320GB SSD, HDD : 6 TB)

Visual Studio Code: 1.75.1

IntelliJ IDEA: 2022.3.1 (Ultimate Edition)

JVM : OpenJDK 11

Docker : 23.0.5

Node.js : 18.15.0

TypeScript: 4.9.5

Redis :

Nginx :

Jenkins :

Azure :

Kubernetes:

@ 2023. A707 갯벌소프트
all right reserved.

2. 설정 파일 목록 환경 변수 정보

Frontend:

- Dockerfile: /front
- .env: /front
- front.conf: /front

Backend_Auth:

- Dockerfile: /back/auth
- application.yml: kubernetes secret
- application.yml: (develop) /back/auth/src/main/resources

Backend_Quiz:

- Dockerfile: /back/quiz
- application.yml: kubernetes secret
- application.yml: (develop) /back/quiz/src/main/resources

Backend_QuizRoom:

- Dockerfile: /back/quizroom
- application.yml: kubernetes secret
- application.yml: (develop) /back/quizroom/src/main/resources

Backend_Grade:

- Dockerfile: /back/grade
- application.yml: kubernetes secret
- application.yml: (develop) /back/grade/src/main/resources

@ 2023. A707 갯벌소프트
all right reserved.

Kubernetes - Control Plane:

```
├── k8s
│   ├── calico.yaml
│   ├── certificate.yaml
│   ├── cluster-issuer.yaml
│   ├── docker-compose.yml
│   ├── ingress-auth.yaml
│   ├── ingress-nginx.yaml
│   ├── ingress-websocket.yaml
│   ├── ingress.yaml
│   ├── kafka_mini.yaml
│   ├── secret.yaml
│   └── zookeeper_mini.yaml
```

Kubernetes - Work Node:

```
.
├── back
│   ├── auth
│   │   ├── application.yml
│   │   └── spring-boot-auth.yaml
│   ├── grade
│   │   ├── application.yml
│   │   └── spring-boot-grade.yaml
│   ├── quiz
│   │   ├── application.yml
│   │   └── sprimg-boot-quiz.yaml
│   ├── quizroom
│   │   ├── application.yml
│   │   └── spring-boot-quizroom.yaml
│   └── user
├── db
│   ├── mariadb-env2.db
│   ├── mariadb-quiz2.yaml
│   ├── redisdb-auth.yaml
│   └── redisdb-session.yaml
├── front
│   └── react-app.yaml
└── k8s
    └── calico.yaml
```

@ 2023. A707 갯벌소프트
all right reserved.

3. 쿠버네티스 설정

Nginx

Ingress-nginx.yaml:

```
curl -o ingress-nginx.yaml
https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.6.
4/deploy/static/provider/baremetal/deploy.yaml

# NodePort에 80, 443 포트 할당
vi ingress-nginx.yaml
```

```
# Service 부분에 nodePort 추가
~~~
ports:
  - appProtocol: http
    name: http
    port: 80
    protocol: TCP
    targetPort: http
    nodePort: 80
  - appProtocol: https
    name: https
    port: 443
    protocol: TCP
    targetPort: https
    nodePort: 443
~~~

# 마스터노드에 생성하려면 nodeSelector에 다음 내용 추가
~~~
spec:
  nodeSelector:
    node-role.kubernetes.io/control-plane: ""
~~~
```

@ 2023. A707 갯벌소프트
all right reserved.

Mariadb

mariadb-env2.db:

```
MYSQL_HOST=%
MYSQL_PORT=3308
MYSQL_ROOT_PASSWORD=ekdrmsehdrms1111!!
MYSQL_DATABASE=luckquiz
MYSQL_USER=carrot707
MYSQL_PASSWORD=ekdrmsehdrms1111!!
```

mariadb-quiz2.yaml:

```yaml
apiVersion: v1
kind: Service
metadata:
  name: mariadb-quiz2
spec:
  type: NodePort
  ports:
    - protocol: TCP
      name: mariadb-quiz2
      port: 3306
      targetPort: 3306
      nodePort: 3308
  selector:
    app: mariadb-quiz2
---
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mariadb-pv2
  labels:
    type: local
spec:
  storageClassName: manual
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/data/mariadb2"
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mariadb-pv-claim2
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
---
```

@ 2023. A707 갯벌소프트
all right reserved.

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mariadb-quiz2
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mariadb-quiz2
  template:
    metadata:
      labels:
        app: mariadb-quiz2
    spec:
      nodeSelector:
       node-role.kubernetes.io/control-plane: "" #create at master node
      containers:
        - image: mariadb:latest
          name: mariadb-quiz2
          envFrom:
            - secretRef:
                name: mariadb-bdg2
          ports:
            - containerPort: 3306
              name: mariadb2
          volumeMounts:
            - name: mariadb-persistent-storage2
              mountPath: /var/lib/mysql
      volumes:
        - name: mariadb-persistent-storage2
          persistentVolumeClaim:
            claimName: mariadb-pv-claim2
```

Redisdb

redisdb-session.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: redisdb-session
spec:
  type: NodePort
  ports:
    - protocol: TCP
      name: redisdb-session
      port: 6379
      targetPort: 6379
      nodePort: 3309
  selector:
    app: redisdb-session
---
```

@ 2023. A707 갯벌소프트
all right reserved.

```yaml
---
apiVersion: v1
kind: PersistentVolume
metadata:
  name: redisdb-pv2
  labels:
    type: local
spec:
  storageClassName: manual
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/data/redisdb/quiz"
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: redisdb-pv-claim2
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
      #allowVolumeExpansion: true
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: redisdb-session
spec:
  replicas: 1
  selector:
    matchLabels:
      app: redisdb-session
  template:
    metadata:
      labels:
        app: redisdb-session
    spec:
      nodeSelector:
              #node-role.kubernetes.io/control-plane: "" #create at master node
      containers:
        - image: redis
          name: redisdb-session
          args: ["--requirepass", "eodrms1111!"]
          ports:
            - containerPort: 6379
              name: redisdb
          volumeMounts:
            - name: redisdb-persistent-storage2
              mountPath: /var/lib/mysql
      volumes:
        - name: redisdb-persistent-storage2
          persistentVolumeClaim:
            claimName: redisdb-pv-claim2
```

@ 2023. A707 갯벌소프트
all right reserved.

kubectl create secret generic spring-boot-{service}-secret -from-file="application.yml" -n {namespace}

AuthServer

Spring-boot-auth.yaml:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: spring-boot-auth
  namespace: spring-boot-v2
spec:
  replicas: 1
  revisionHistoryLimit: 2
  selector:
    matchLabels:
      app: spring-boot-auth
  template:
    metadata:
      labels:
        app: spring-boot-auth
    spec:
      containers:
      - name: spring-boot-auth
        image: docker.io/carrot707/luckquiz:spring-boot-auth-2
        ports:
        - containerPort: 8080
        volumeMounts:
        - name: secret-volume
          mountPath: /config/secret
          readOnly: true
        env:
        - name: SPRING_CONFIG_LOCATION
          value: "file:/config/secret/application.yml"
      volumes:
      - name: secret-volume
        secret:
          secretName: spring-boot-auth-secret
      imagePullSecrets:
      - name: dockerhub-secret

---
```

@ 2023. A707 갯벌소프트
all right reserved.

```yaml
---
apiVersion: v1
kind: Service
metadata:
  name: spring-boot-auth
  namespace: spring-boot-v2
spec:
  selector:
    app: spring-boot-auth
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
  type: ClusterIP
---
apiVersion: v1
kind: Service
metadata:
  name: auth-external-v2
  namespace: default
spec:
  type: ExternalName
  externalName: spring-boot-auth.spring-boot-v2.svc.cluster.local
  ports:
    - name: http
      port: 8080
```

@ 2023. A707 갯벌소프트
all right reserved.

GradeServer

Spring-boot-grade.yaml:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: spring-boot-grade
  namespace: spring-boot-v2
spec:
  replicas: 1
  revisionHistoryLimit: 2
  selector:
    matchLabels:
      app: spring-boot-grade
  template:
    metadata:
      labels:
        app: spring-boot-grade
    spec:
      containers:
      - name: spring-boot-grade
        image: docker.io/carrot707/luckquiz:spring-boot-grade-4
        ports:
        - containerPort: 8080
        volumeMounts:
        - name: secret-volume
          mountPath: /config/secret
          readOnly: true
        env:
        - name: SPRING_CONFIG_LOCATION
          value: "file:/config/secret/application.yml"
      volumes:
      - name: secret-volume
        secret:
          secretName: spring-boot-grade-secret
      imagePullSecrets:
      - name: dockerhub-secret

---
```

@ 2023. A707 갯벌소프트
all right reserved.

```yaml
---
apiVersion: v1
kind: Service
metadata:
  name: spring-boot-grade
  namespace: spring-boot-v2
spec:
  selector:
    app: spring-boot-grade
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
  type: ClusterIP
---
apiVersion: v1
kind: Service
metadata:
  name: grade-external-v2
  namespace: default
spec:
  type: ExternalName
  externalName: spring-boot-grade.spring-boot-v2.svc.cluster.local
  ports:
    - name: http
      port: 8080
```

@ 2023. A707 갯벌소프트
all right reserved.

QuizServer

Spring-boot-quiz.yaml:

```yaml
metadata:
  name: spring-boot-quiz
  namespace: spring-boot-v2
spec:
  replicas: 1
  revisionHistoryLimit: 2
  selector:
    matchLabels:
      app: spring-boot-quiz
  template:
    metadata:
      labels:
        app: spring-boot-quiz
    spec:
      containers:
      - name: spring-boot-quiz
        image: docker.io/carrot707/luckquiz:spring-boot-quiz-28
        ports:
        - containerPort: 8080
        volumeMounts:
        - name: secret-volume
          mountPath: /config/secret
          readOnly: true
        env:
        - name: SPRING_CONFIG_LOCATION
          value: "file:/config/secret/application.yml"
      volumes:
      - name: secret-volume
        secret:
          secretName: spring-boot-quiz-secret
      imagePullSecrets:
      - name: dockerhub-secret
```

@ 2023. A707 갯벌소프트
all right reserved.

```yaml
---
apiVersion: v1
kind: Service
metadata:
  name: spring-boot-quiz
  namespace: spring-boot-v2
spec:
  selector:
    app: spring-boot-quiz
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
  type: ClusterIP
---
apiVersion: v1
kind: Service
metadata:
  name: quiz-external-v2
  namespace: default
spec:
  type: ExternalName
  externalName: spring-boot-quiz.spring-boot-v2.svc.cluster.local
  ports:
    - name: http
      port: 8080
```

@ 2023. A707 갯벌소프트
all right reserved.

QuizRoomServer

Spring-boot-quizroom.yaml:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: spring-boot-quizroom
  namespace: spring-boot-v2
spec:
  replicas: 1
  revisionHistoryLimit: 2
  selector:
    matchLabels:
      app: spring-boot-quizroom
  template:
    metadata:
      labels:
        app: spring-boot-quizroom
    spec:
      containers:
      - name: spring-boot-quizroom
        image: docker.io/carrot707/luckquiz:spring-boot-quizroom-1
        ports:
        - containerPort: 8080
        volumeMounts:
        - name: secret-volume
          mountPath: /config/secret
          readOnly: true
        env:
        - name: SPRING_CONFIG_LOCATION
          value: "file:/config/secret/application.yml"
      volumes:
      - name: secret-volume
        secret:
          secretName: spring-boot-quizroom-secret
      imagePullSecrets:
      - name: dockerhub-secret

---
```

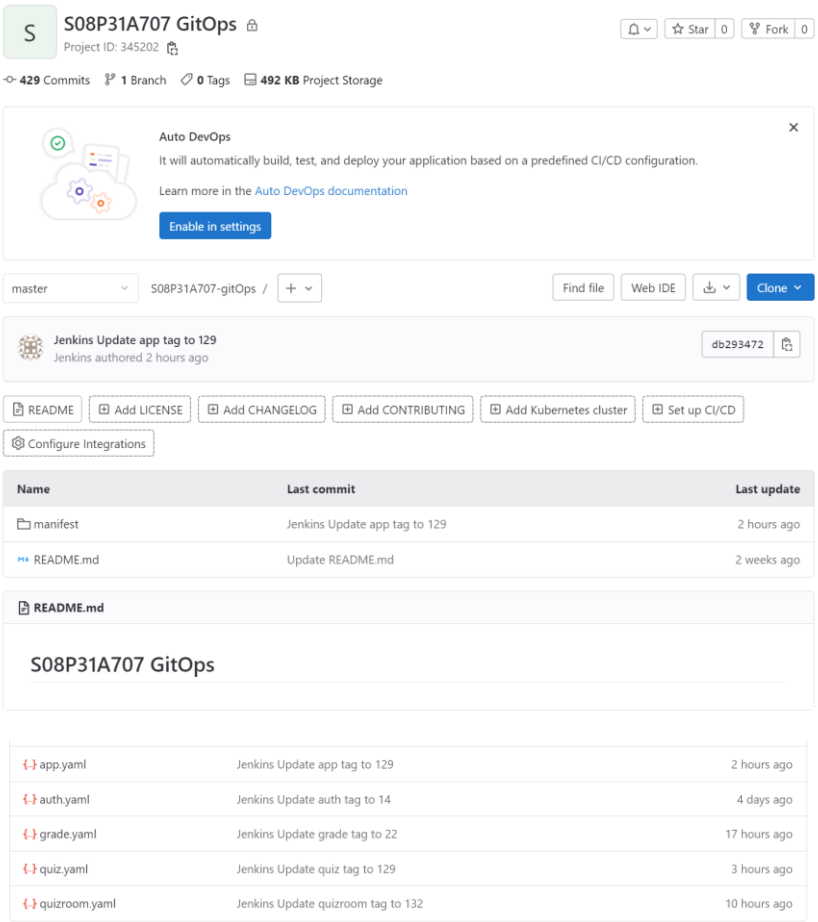
@ 2023. A707 갯벌소프트
all right reserved.

```yaml
---
apiVersion: v1
kind: Service
metadata:
  name: spring-boot-quizroom
  namespace: spring-boot-v2
spec:
  selector:
    app: spring-boot-quizroom
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
  type: ClusterIP
---
apiVersion: v1
kind: Service
metadata:
  name: quizroom-external-v2
  namespace: default
spec:
  type: ExternalName
  externalName: spring-boot-quizroom.spring-boot-v2.svc.cluster.local
  ports:
    - name: http
      port: 8080
```

@ 2023. A707 갯벌소프트
all right reserved.

3. GitOps 및 ArgoCD

GitOps



@ 2023. A707 갯벌소프트
all right reserved.

3. GitOps 및 ArgoCD

app.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: react-app
  namespace: react-v2
spec:
  replicas: 1
  revisionHistoryLimit: 2
  selector:
    matchLabels:
      app: react-app
  template:
    metadata:
      labels:
        app: react-app
    spec:
      containers:
      - name: react-app
        image: docker.io/carrot707/luckquiz:react-app-129
        ports:
        - containerPort: 3000
      imagePullSecrets:
      - name: dockerhub-secret
```

@ 2023. A707 갯벌소프트
all right reserved.

auth.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: spring-boot-auth
  namespace: spring-boot-v2
spec:
  replicas: 1
  revisionHistoryLimit: 2
  selector:
    matchLabels:
      app: spring-boot-auth
  template:
    metadata:
      labels:
        app: spring-boot-auth
    spec:
      containers:
      - name: spring-boot-auth
        image: docker.io/carrot707/luckquiz:spring-boot-auth-14
        ports:
        - containerPort: 8080
        volumeMounts:
        - name: secret-volume
          mountPath: /config/secret
          readOnly: true
        env:
        - name: SPRING_CONFIG_LOCATION
          value: "file:/config/secret/application.yml"
      volumes:
      - name: secret-volume
        secret:
          secretName: spring-boot-auth-secret
      imagePullSecrets:
      - name: dockerhub-secret
```

@ 2023. A707 갯벌소프트
all right reserved.

3. 카프카 설정

Apache Kafa 다운로드

```
wget https://dlcdn.apache.org/kafka/3.4.0/kafka_2.13-3.4.0.tgz
```

압축 해제

```
tar -xzf kafka_2.13-3.4.0.tgz
```

kafka_2.13-3.4.0/config/zookeeper.properties

```
// 1번 서버 zookeeper 설정
dataDir=/tmp/zookeeper
clientPort=3181
maxClientCnxns=0
admin.enableServer=false
tickTime=2000
initLimit=20
syncLimit=5
electionPort=3288
quorumPort=3888
server.1=0.0.0.0:3288:3888
server.2=2번서버IP:3288:3888
server.3=3번서버IP:3288:3888
```

@ 2023. A707 갯벌소프트
all right reserved.

kafka_2.13-3.4.0/config/zookeeper.properties

```
// 2번 서버 zookeeper 설정
clientPort=3181
maxClientCnxns=0
admin.enableServer=false
tickTime=2000
initLimit=20
syncLimit=5
electionPort=3288
quorumPort=3888
server.1=1번서버IP:3288:3888
server.2=0.0.0.0:3288:3888
server.3=3번서버IP:3288:3888
```

```
// 3번 서버 zookeeper 설정
dataDir=/tmp/zookeeper
clientPort=3181
maxClientCnxns=0
admin.enableServer=false
tickTime=2000
initLimit=20
syncLimit=5
electionPort=3288
quorumPort=3888
server.1=1번서버IP:3288:3888
server.2=2번서버IP:3288:3888
server.3=0.0.0.0:3288:3888
```

@ 2023. A707 갯벌소프트
all right reserved.

kafka_2.13-3.4.0/config/server.properties

```
// 1번 서버 kafka broker 설정
listeners=PLAINTEXT://:9092
broker.id=0
advertised.listeners=PLAINTEXT://1번서버IP:9092
num.network.threads=3
num.io.threads=8
socket.send.buffer.bytes=102400
socket.receive.buffer.bytes=102400
socket.request.max.bytes=104857600
############################# Log Basics #############################
log.dirs=/tmp/kafka-logs
num.partitions=1
num.recovery.threads.per.data.dir=1
############################# Internal Topic Settings  #############################
offsets.topic.replication.factor=1
transaction.state.log.replication.factor=1
transaction.state.log.min.isr=1
############################# Log Retention Policy #############################
log.retention.hours=168
log.retention.check.interval.ms=300000
############################# Zookeeper #############################
zookeeper.connect=0.0.0.0:3181,2번서버IP:3181,3번서버IP:3181
zookeeper.connection.timeout.ms=18000
############################# Group Coordinator Settings #############################
group.initial.rebalance.delay.ms=0

timezone=Asia/Seoul
```

```
// 2번 서버 kafka broker 설정
listeners=PLAINTEXT://:9092
broker.id=1
advertised.listeners=PLAINTEXT://2번서버IP:9092
num.network.threads=3
num.io.threads=8
socket.send.buffer.bytes=102400
socket.receive.buffer.bytes=102400
socket.request.max.bytes=104857600
############################# Log Basics #############################
log.dirs=/tmp/kafka-logs
num.partitions=1
num.recovery.threads.per.data.dir=1
############################# Internal Topic Settings  #############################
offsets.topic.replication.factor=1
transaction.state.log.replication.factor=1
transaction.state.log.min.isr=1
############################# Log Retention Policy #############################
log.retention.hours=168
log.retention.check.interval.ms=300000
############################# Zookeeper #############################
zookeeper.connect=1번서버IP:3181,0.0.0.0:3181,3번서버IP:3181
zookeeper.connection.timeout.ms=18000
############################# Group Coordinator Settings #############################
group.initial.rebalance.delay.ms=0

timezone=Asia/Seoul
```

@ 2023. A707 갯벌소프트
all right reserved.

```
// 3번 서버 kafka broker 설정
listeners=PLAINTEXT://:9092
broker.id=2
advertised.listeners=PLAINTEXT://3번서버IP:9092
num.network.threads=3
num.io.threads=8
socket.send.buffer.bytes=102400
socket.receive.buffer.bytes=102400
socket.request.max.bytes=104857600
############################# Log Basics #############################
log.dirs=/tmp/kafka-logs
num.partitions=1
num.recovery.threads.per.data.dir=1
############################# Internal Topic Settings  #############################
offsets.topic.replication.factor=1
transaction.state.log.replication.factor=1
transaction.state.log.min.isr=1
############################# Log Retention Policy #############################
log.retention.hours=168
log.retention.check.interval.ms=300000
############################# Zookeeper #############################
zookeeper.connect=1번서버IP:3181,2번서버IP:3181,0.0.0.0:3181
zookeeper.connection.timeout.ms=18000
############################# Group Coordinator Settings #############################
group.initial.rebalance.delay.ms=0

timezone=Asia/Seoul
```

주키퍼 백그라운드 실행

```
nohup kafka_2.13-3.4.0/bin/zookeeper-server-start.sh config/zookeeper.properties
```

카프카 백그라운드 실행

```
nohup kafka_2.13-3.4.0/bin/kafka-server-start.sh config/server.properties
```

@ 2023. A707 갯벌소프트
all right reserved.

kafdrop 설치 및 실행

```yaml
# kafdrop 설정
version: '3'
services:
  kafdrop:
    container_name: kafdrop
    image: obsidiandynamics/kafdrop
    restart: "no"
    ports:
      - "9000:9000"
    environment:
      KAFKA_BROKER_CONNECT: "1번서버IP:9092"
      SERVER_PORT: 9000
      MANAGEMENT_SERVER_PORT: 9000
```

@ 2023. A707 갯벌소프트
all right reserved.

4. Frontend 배포

front.cof

```
server {
listen 3000;
    location / {
    root /app/build;
    index index.html;
    try_files $uri $uri/ /index.html;
    }
}
```

Dockerfile

```
FROM nginx:stable-alpine
WORKDIR /app
RUN mkdir ./build
ADD ./build ./build
RUN rm /etc/nginx/conf.d/default.conf
COPY ./front.conf /etc/nginx/conf.d
EXPOSE 3000
CMD ["nginx", "-g", "daemon off;"]
```

@ 2023. A707 갯벌소프트
all right reserved.

4. Backend 배포

```
FROM openjdk:11

WORKDIR /app

ARG JAR_FILE=build/libs/auth-0.0.1-SNAPSHOT.jar

COPY ${JAR_FILE} app.jar

EXPOSE 8080

ENTRYPOINT ["java","-jar", "-Duser.timezone=Asia/Seoul", "-Dspring.config.location=${SPRING_CONFIG_LOCATION}", "app.jar"]
```

@ 2023. A707 갯벌소프트
all right reserved.

5. Jenkins

S	W	Name ↓	최근 성공	최근 실패	최근 소요 시간
✓	☼	client_front	3 days 2 hr #129	3 days 11 hr #117	1 min 16 sec
✓	☼	consumer_test	25 days #9	26 days #2	14 sec
✓	☼	producer_test	22 days #35	27 days #4	13 sec
✓	☼	server_auth	7 days 21 hr #14	13 days #7	35 sec
✓	☼	server_grade	3 days 17 hr #22	20 days #3	43 sec
✓	☼	server_quiz	3 days 3 hr #129	4 days 19 hr #93	1 min 9 sec
✓	☼	server_quizroom	3 days 10 hr #132	7 days 2 hr #85	48 sec

Build Triggers

- [ ] Build after other projects are built ?
- [ ] Build periodically ?
- [x] Build when a change is pushed to GitLab. GitLab webhook URL: http://k8a707.p.ssafy.io:8080/project/server_quiz ?

Enabled GitLab triggers

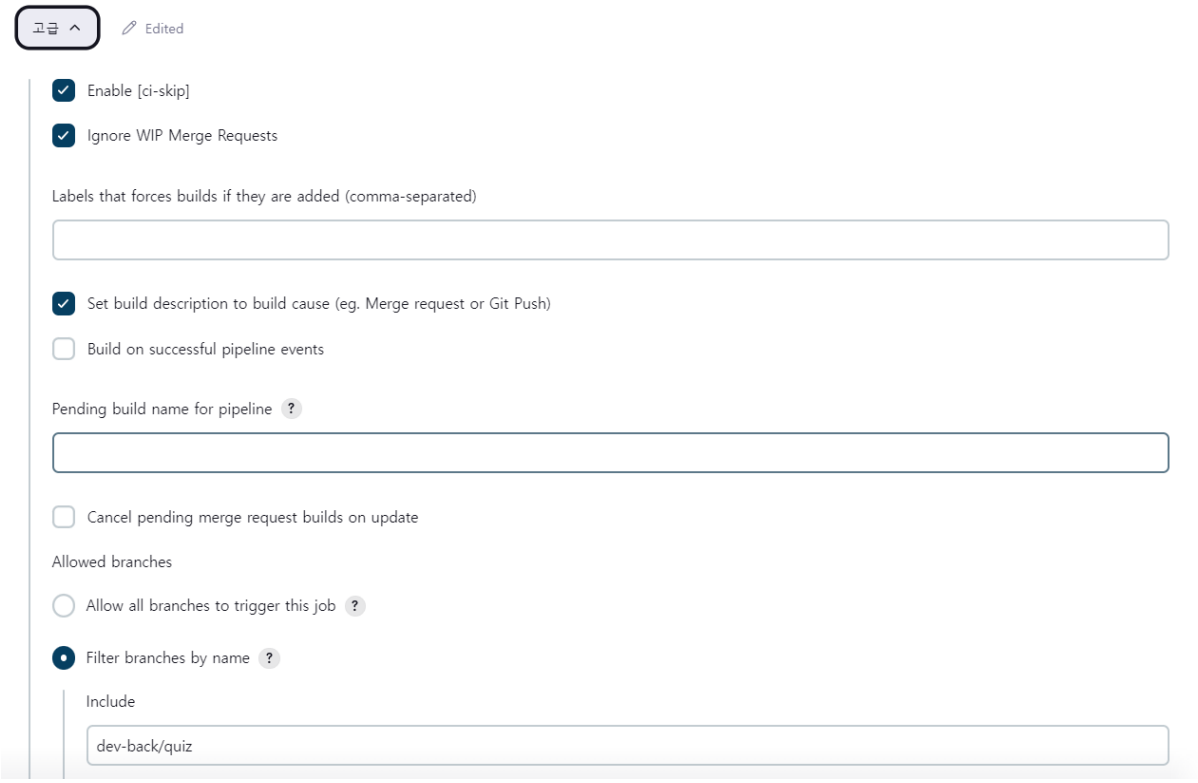
- [ ] Push Events
- [ ] Push Events in case of branch delete
- [ ] Opened Merge Request Events
- [ ] Build only if new commits were pushed to Merge Request ?
- [x] Accepted Merge Request Events
- [ ] Closed Merge Request Events

Rebuild open Merge Requests

Never

- [x] Approved Merge Requests (EE-only)
- [x] Comments

@ 2023. A707 갯벌소프트
all right reserved.



+ 시크릿토큰 추가하기

```
pipeline {
    agent any
    environment {
        DOCKER_REPOSITORY = "carrot707/luckquiz"
        DOCKERHUB_CREDENTIALS = credentials('dockerhub-username-and-password')
        GITLAB_CREDENTIALS = credentials('gitlab-username-and-password')
    }
    tools {
        nodejs "nodejs"
    }

    stages {
        stage('Clone Git Repository') {
            steps {
              git credentialsId: 'gitlab-username-and-password',
              branch: "dev-front",
              url: 'https://lab.ssafy.com/s08-final/S08P31A707.git'
            }
        }

        stage('Build React Project') {
            steps{
                echo 'Frontend Project Build Step'
                dir('front') {
                    sh 'npm install'
                    sh 'npm run build'
                }
            }
        }
```

@ 2023. A707 갯벌소프트
all right reserved.

```
    stage('Signin Dockerhub') {
        steps {
            sh """
            echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin
            """
        }
    }

    stage('Push') {
        steps {
            sh """ docker push $DOCKER_REPOSITORY:react-app-${env.BUILD_NUMBER}"""
        }
    }

    stage('Deploy') {
        steps {
            dir('git') {
                withCredentials([usernamePassword(credentialsId: 'gitlab-username-and-password',
                usernameVariable: 'GITLAB_USERNAME', passwordVariable: 'GITLAB_PASSWORD')]) {
                    git credentialsId: 'gitlab-username-and-password', branch: "master",
                    url: 'https://lab.ssafy.com/dodamond222/S08P31A707-gitOps.git'
                    sh """
                    sed -i 's/luckquiz:react-app-\\([^:]*\\)/luckquiz:react-app-${env.BUILD_NUMBER}/g' manifest/app.yaml
                    git add manifest/app.yaml
                    git commit -m 'Jenkins Update app tag to ${env.
                    git push https://dodamond222:${GITLAB_PASSWORD}@lab.ssafy.com/dodamond222/S08P31A707-gitOps.git
                    """
                }
            }
        }
    }
}
```

Follow link (ctrl + click)

@ 2023. A707 갯벌소프트
all right reserved.