

K8S Demo Project

Organizing Frontend/Backend/Web/DB

모바일시스템공학과 조민욱

adrd1820@gmail.com

K8S

- React deployment, service
 - Node deployment, service
 - MySQL deployment, service
 - DB PV
 - DB PVC
 - DB SC
-
- 필수적인 실제 클러스터 구성 YAML 요소
 - YAML파일 별도 첨부... 확인

K8S

- 이전에 설정한 주소를 그대로 사용 불가
- K8S에 맞게 서비스 이름으로 URL을 호출
- 이렇게 진행하지 않으면 나중에 Backend서버에서 DB를 찾을 수 없습니다.
- 클러스터 내에서는 URL 대신 서비스 이름을 사용합니다.

```
const mysql = require('mysql2');

const db = mysql.createConnection({
  host: 'mysql-service', // host: 'localhost', host: '10.0.20.71',
  port: 3306,
  user: 'root',
  password: 'rootpassword',
  database: 'myappdb',
});
```

db.js

```
import axios from 'axios';

const api = axios.create({
  baseURL: 'http://node-app-service:80',
  timeout: 10000, // 요청 제한 시간 설정
  headers: {
    'Content-Type': 'application/json',
  },
});

export default api;
```

Api.jsx

K8S

```
volumeMounts:
  - name: mysql-persistent-storage
    mountPath: /var/lib/mysql
volumes:
  - name: mysql-persistent-storage
    persistentVolumeClaim:
      claimName: mysql-pvc
```

MySQL

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 0.1Gi
  storageClassName: mysql-storage-class
```

PVC

요청 용량이 PV 자체
용량보다 크면 안됩니다.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mysql-pv
spec:
  capacity:
    storage: 0.1Gi
  accessModes:
    - ReadWriteOnce
  storageClassName: mysql-storage-class
  hostPath:
    path: /home/minuk/react_db
  nodeAffinity:
    required:
      nodeSelectorTerms:
        - matchExpressions:
            - key: kubernetes.io/hostname
              operator: In
              values:
                - muchoubuntu
```

PV

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: mysql-storage-class
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: WaitForFirstConsumer
```

SC

로컬 볼륨 사용
지정합니다.

Node Name(VM Name)의
해당 디렉토리에 위치합니다.

K8S

- 데이터베이스가 돌아가는지 확인
 - Phpmyadmin 이용 → 노드 포트를 이용해 테스트 접속
 - `http://10.0.20.71:30002/`
 - 사용자명 : root
 - 암호 : rootpassword
- 현재까지 완료된 것...
 - Node 배포 확인
 - DB 배포 확인
 - Node-DB 연동 확인... 이제 GO!

The screenshot shows the phpMyAdmin interface for a MySQL database named 'myappdb'. The selected table is 'users'. The SQL query executed is 'SELECT * FROM `users`'. The result shows 2 rows of data. The interface includes a top navigation bar with options like '보기' (View), '구조' (Structure), 'SQL', '검색' (Search), '삽입' (Insert), and '내보내기' (Export). Below the query, there are checkboxes for '프로파일링' (Profiling), '인라인 편집' (Inline edit), '수정' (Edit), 'SQL 해석' (SQL explain), 'PHP 코드 보기' (Show PHP code), and '다시 보기' (Refresh). The '모두 보기' (Show all) checkbox is checked, and the '행 갯수' (Number of rows) is set to 25. The '행 필터링' (Row filtering) is set to '현재 테이블 검색' (Search current table). Below the query, there is an 'Extra options' button. The table data is displayed with columns: id, username, password, and score. The first row has id 1, username 'user1', password 'password1', and score 100. The second row has id 2, username 'user2', password 'password2', and score 150. Each row has checkboxes for '수정' (Edit), '복사' (Copy), and '삭제' (Delete).

	id	username	password	score
<input type="checkbox"/> 수정 <input type="checkbox"/> 복사 <input type="checkbox"/> 삭제	1	user1	password1	100
<input type="checkbox"/> 수정 <input type="checkbox"/> 복사 <input type="checkbox"/> 삭제	2	user2	password2	150

K8S

- Backend는 나중에 DB가 생긴 뒤 접속을 시도해야 합니다.
- 즉 DB 관련 YAML 파일 먼저 배포해야 합니다.
- DB와 Backend는 성공적으로 연결되었지만 React가 정상 작동하지 않네요.
 - 개발자 도구에서 해당 에러를 확인할 수 있습니다.
 - 프론트에서 말썽인데... 일단 아예 Node 쪽으로 트래픽 전송 자체가 안되네요.
 - WebSocketClient.js:13 WebSocket connection to 'ws://10.0.20.71:3000/ws' failed:

```
minuk@muchoubuntu:~/react_app/k8s$ kubectl logs node-app-deployment-f87fdfdb6-2tmlf
Server is running on http://localhost:5000
Connected to MySQL database
```

```
✖ ▼ WebSocket connection to 'ws://10.0.20.71:3000/ws' failed:
  WebSocketClient @ WebSocketClient.js:13
    initSocket      @ socket.js:27
    (anonymous)     @ socket.js:51
  Show logs
```

```
✖ ▶ WebSocket connection to 'ws://10.0.20.71:3000/ws' failed:
```

```
✖ ▶ WebSocket connection to 'ws://10.0.20.71:3000/ws' failed:
```

```
✖ ▶ WebSocket connection to 'ws://10.0.20.71:3000/ws' failed:
```

K8S

- Dockerfile을 살짝 변경해봤지만, 역시 안됩니다.
- 찾아보기론 HTTP와 별개인, 양방향 TCP를 활용하는 방식을 Web Socket이라고 합니다.
- localhost:3000을 통한 소켓 연결이 불가능한 상황입니다.
 - HTTP로는 localhost:30000을 통해 노드 포트로 접속...
 - Web Socket으로는 localhost:3000을 통해 접속 시도...
- npm install express ws
 - 그런데 생각해보니 현재 앱은 Socket 자체를 사용하지 않아요.
 - 소켓이 연결되어 있지 않아도 http 통신을 통해 데이터 교환하면 됩니다.
 - 다시 한번 자세히 살펴봅시다.

```
FROM ubuntu

RUN apt-get update && \
    apt-get install -y npm && \
    npm cache clean --force

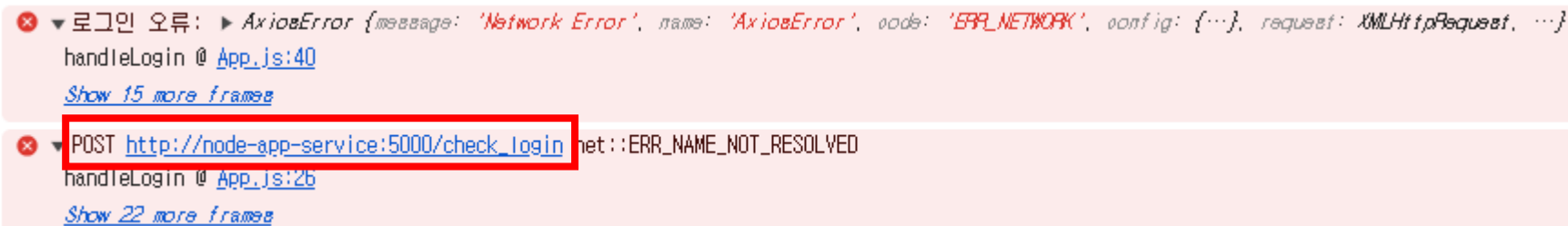
RUN mkdir /app
ADD . /app
WORKDIR /app

RUN npm install http-proxy-middleware && \
    npm install react-router-dom && \
    npm install axios

ENV WDS_SOCKET_PORT=0

EXPOSE 3000
CMD ["npm", "start"]
```

K8S



소켓을 무시하더라도, 일단 HTTP를 이용한 통신은 가능해야 하는데...

서비스 포트가 80번으로 열려 있는데 갑자기 5000번 포트로 POST를 시도?

```
import axios from 'axios';

const api = axios.create({
  baseURL: 'http://node-app-service:80',
  timeout: 10000, // 요청 제한 시간 설정
  headers: {
    'Content-Type': 'application/json',
  },
});

export default api;
```

Api.jsx

Load Balancer를 활용해 Nginx를 돌리고 포트를 명시해볼까?

K8S

```
# 모든 요청을 react-app-service로 프록시
location / {
    proxy_pass http://react-app-service:80/;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header Origin "";
}
location /api/ {
    proxy_pass http://node-app-service:80/;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header Origin "";
}
```

10.0.20.75

USERNAME

PASSWORD

LOGIN

10.0.20.75/api/

Hello, World!

이 문제가 아닌, 서비스 포트 자체 문제인 걸로...

K8S

- 서비스 포트 자동 변환 문제?
 - <http://node-app-service:80/>
 - 컨테이너 포트에 의해 <http://node-app-service:5000/>으로 자동 변환
 - 배포할 때 5000번 포트, API 포트, Nginx proxy pass 변경해도 안되네요...
- 혹은 CORS 문제?
 - 현재 클러스터 내부에서 서버를 호출하는데?
 - Backend 자체로 접근을 못하는 중
- 이해결...
 - 이미지 → 컨테이너 성공
 - 이미지 → K8S 실패
 - Frontend, Backend, DB, Backend → DB 성공
 - Frontend → DB 실패

DNS

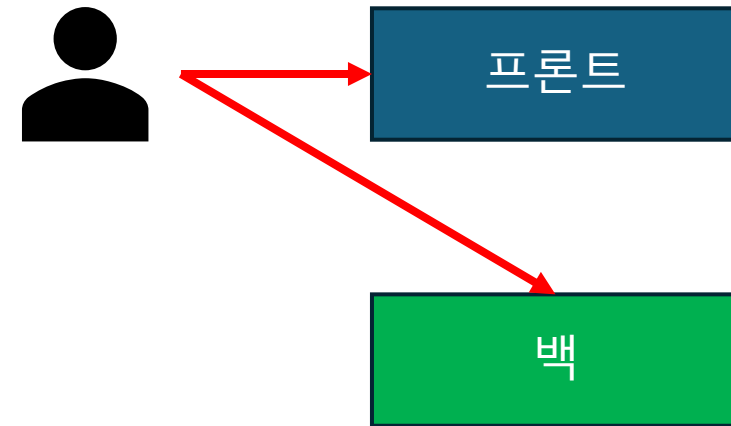
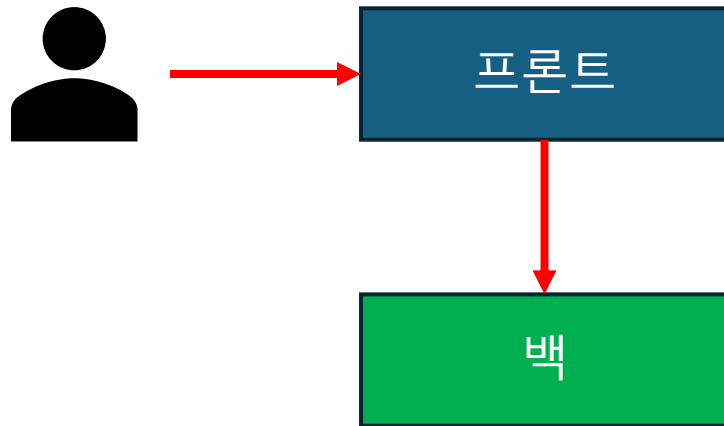
- `kubectl get po -n kube-system -l k8s-app=kube-dns`
- `kubectl get svc -n kube-system -l k8s-app=kube-dns`
- DNS가 node-app-service를 해독하지 못하는 건가?

```
minuk@muchoubuntu:~/react_app/k8s$ kubectl get svc -n kube-system -l k8s-app=kube-dns
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
kube-dns     ClusterIP   10.96.0.10    <none>        53/UDP,53/TCP,9153/TCP  9d
minuk@muchoubuntu:~/react_app/k8s$ kubectl get po -n kube-system -l k8s-app=kube-dns
NAME                                READY   STATUS    RESTARTS   AGE
coredns-57575c5f89-5k6ff            1/1     Running   0           9d
coredns-57575c5f89-zdccv            1/1     Running   0           9d
```

DNS

- 정답

- 프론트에서 백으로 바로 트래픽을 보내지 않고, 사용자가 백으로 접근하는 느낌...
- Middle proxy ware 사용!

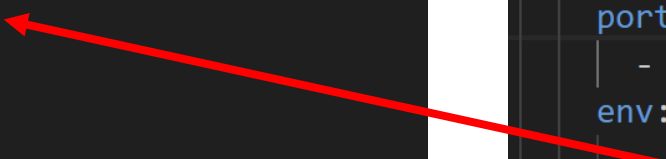


Error : Proxy middleware

```
const { createProxyMiddleware } = require('http-proxy-middleware');

module.exports = function (app) {
  app.use(
    '/api',
    createProxyMiddleware({
      target: process.env.REACT_APP_API_URL,
      changeOrigin: true,
      pathRewrite: {
        '^/api': '',
      }
    })
  );
};
```

```
containers:
  - name: react-app
    image: adaliv/k8s_react:1.5
    ports:
      - containerPort: 3000
    env:
      name: REACT_APP_API_URL
      value: "http://node-app-service:5000"
```



Error : DB connection

프론트 엔드에서 백 엔드 오류가 발생해도
해당 오류일 가능성 참고...

```
const mysql = require('mysql2');

const db = mysql.createPool({
  host: 'mysql-service',
  port: 3306,
  user: 'root',
  password: 'rootpassword',
  database: 'myappdb',
});

db.getConnection((err, connection) => {
  if (err) {
    console.error('Error connecting to MySQL database:', err);
    return;
  }
  console.log('Connected to MySQL database');

  connection.release();
});

module.exports = db;
```

Back에서 DB 연결 성공 후 바로 끊긴다면 MySQL의 고질적인 문제...

다음과 같이 코드 고치기!