

# 나도 해보자! 표준프레임워크 샘플게시판 활용하기

---



# 목 차

---

1. 발표자 소개
2. 샘플게시판 이해
3. 표준프레임워크 버전 업그레이드
4. Spring Boot 전환
5. Gradle 전환
6. Kubernetes 배포
7. 참고자료

# 1. 발표자 소개

❑ 전자정부 표준프레임워크 구축/유지보수 사업참여(2011 ~ 2015)

❑ 사내 Cloud 프로젝트 다수 구축(2017 ~2020)

❑ 사내 개발자 대회 수상(2018)

❑ 사내 Cloud Native 기반 BigData Platform 개발(2022~)

❑ <나도 해보자! 시리즈> 오픈커뮤니티 세미나 발표

- 나도 해보자! 표준프레임워크 개발환경 구축
- 나도 해보자! 표준프레임워크 3.X 업그레이드
- 나도 해보자! Cloud Project with Kubernetes 등 다수

❑ 슈퍼개발자K 시즌3 동상 수상(2014)

❑ 행정안전부장관 수상(2021)

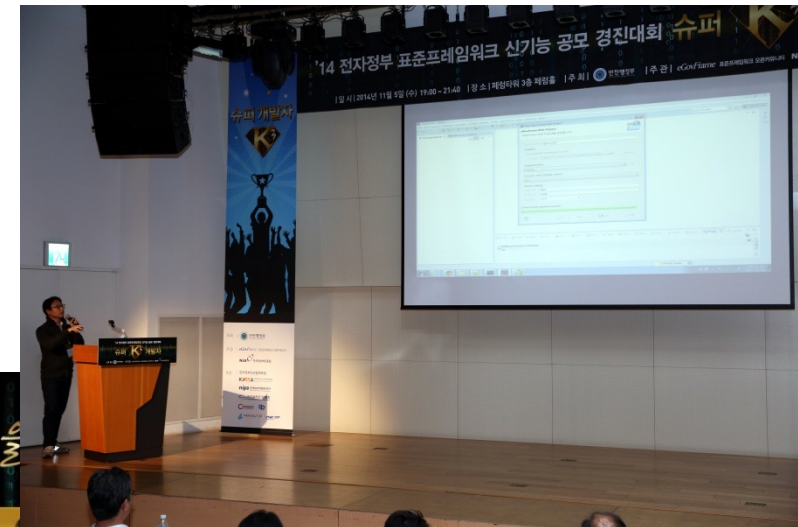
❑ 오픈플랫폼(PaaS) 전문가과정 강의(2016)

- 강사(허광남 등 9명)

❑ 現오픈커뮤니티 리더(2015 ~)

❑ 現OPA(Open Platform Alliance) 리더(2022~)

표준프레임워크 오픈커뮤니티  
**eGovFrame**



## 2. 샘플게시판 이해

### □ 간단한 CRUD만 있는 게시판

List Sample

Name

검색

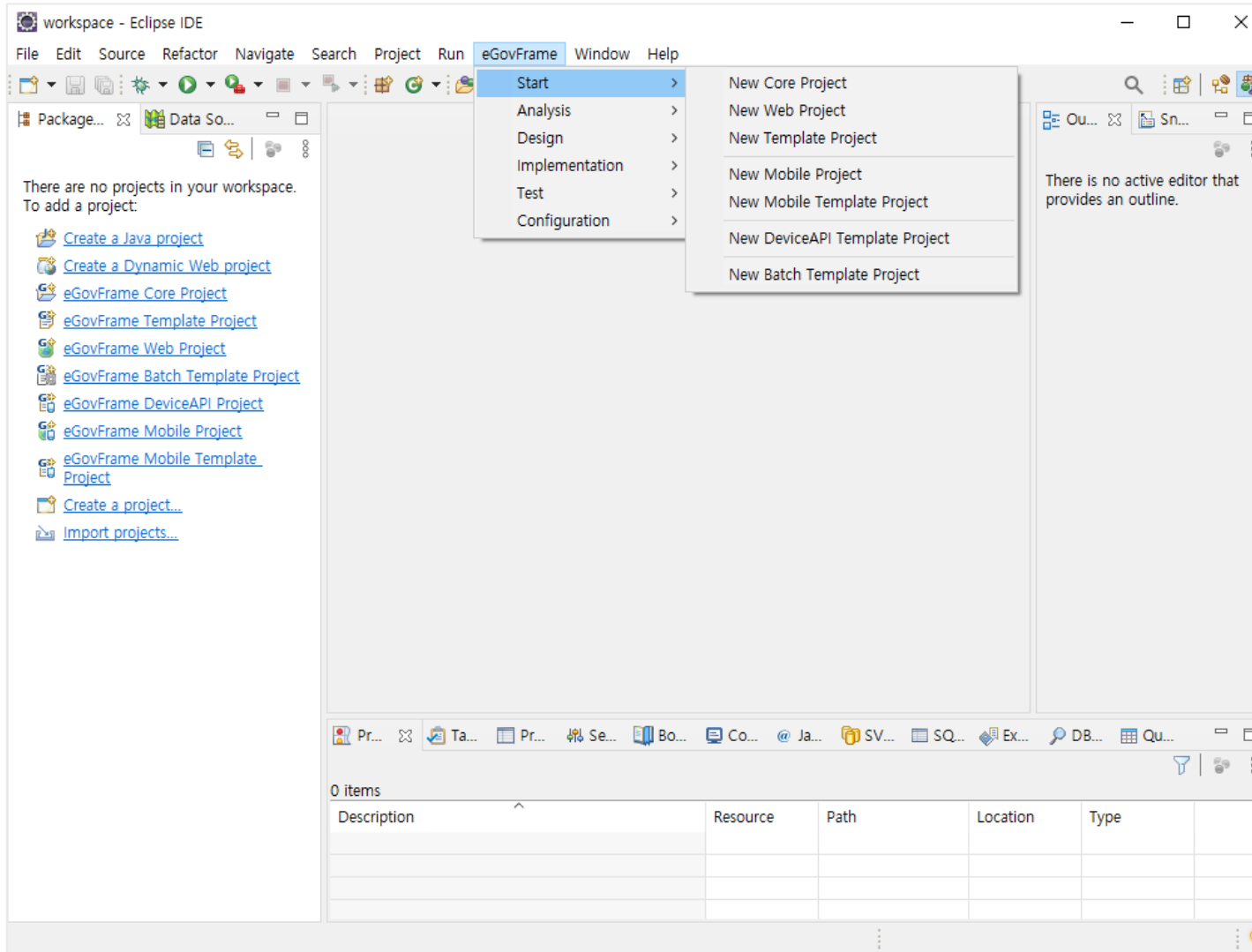
| No  | 카테고리ID       | 카테고리명               | 사용여부 | 설명                | 등록자  |
|-----|--------------|---------------------|------|-------------------|------|
| 114 | SAMPLE-00114 | Runtime Environment | Y    | Integration Layer | eGov |
| 113 | SAMPLE-00113 | Runtime Environment | Y    | Integration Layer | eGov |
| 112 | SAMPLE-00112 | Runtime Environment | Y    | Integration Layer | eGov |
| 111 | SAMPLE-00111 | Runtime Environment | Y    | Integration Layer | eGov |
| 110 | SAMPLE-00110 | Runtime Environment | Y    | Integration Layer | eGov |
| 109 | SAMPLE-00109 | Runtime Environment | Y    | Integration Layer | eGov |
| 108 | SAMPLE-00108 | Runtime Environment | Y    | Integration Layer | eGov |
| 107 | SAMPLE-00107 | Runtime Environment | Y    | Integration Layer | eGov |
| 106 | SAMPLE-00106 | Runtime Environment | Y    | Integration Layer | eGov |
| 105 | SAMPLE-00105 | Runtime Environment | Y    | Integration Layer | eGov |

<< < 1 2 3 4 5 6 7 8 9 10 > >>

등록

## 2. 샘플게시판 이해

### □ 게시판 소스 생성



## 2. 샘플게시판 이해

### □ 게시판 소스 생성

New eGovFrame Web Project

**eGovFrame Web Project**  
eGovFrame Web Project를 생성합니다.

Project name:

Contents

☒ Use default Workspace location

Location:

Target Runtime

Dynamic Web Module version

Maven Setting

Group Id:

Artifact Id:

Version:



New eGovFrame Web Project

**Generate Example**  
예제 소스를 생성하시려면 아래 선택 버튼을 체크 하십시오.

☒ Generate Example

Files

|   |                                  |
|---|----------------------------------|
| <input type="checkbox"/> index.jsp                | src/main/webapp                  |
| <input type="checkbox"/> title_dot.gif            | src/main/webapp/images/egovfr... |
| <input type="checkbox"/> th_bg.gif                | src/main/webapp/images/egovfr... |
| <input type="checkbox"/> paging_line.gif          | src/main/webapp/images/egovfr... |
| <input type="checkbox"/> civilappeal_topmn_bg.jpg | src/main/webapp/images/egovfr... |
| <input type="checkbox"/> btn_bg_r.gif             | src/main/webapp/images/egovfr... |
| <input type="checkbox"/> btn_bg_l.gif             | src/main/webapp/images/egovfr... |
| <input type="checkbox"/> btn_page_pre10.gif       | src/main/webapp/images/egovfr... |
| <input type="checkbox"/> btn_page_pre1.gif        | src/main/webapp/images/egovfr... |
| <input type="checkbox"/> btn_page_next10.gif      | src/main/webapp/images/egovfr... |
| <input type="checkbox"/> btn_page_next1.gif       | src/main/webapp/images/egovfr... |
| <input type="checkbox"/> sample.css               | src/main/webapp/css/egovframe... |
| <input type="checkbox"/> error.jsp                | src/main/webapp/common           |
| <input type="checkbox"/> web.xml                  | src/main/webapp/WEB-INF          |

## 2. 샘플게시판 이해

### ❑ Maven Repository 에러

m pom.xml ~/Documents/IdeaProjects/opdc-seminar/sample-3\_10\_0 33개의 문제

- ❗ 종속성 'egovframework.rte:egovframework.rte.ptl.mvc:3.10.0'을(를) 찾을 수 없습니다 :72
- ❗ 종속성 'egovframework.rte:egovframework.rte.psl.dataaccess:3.10.0'을(를) 찾을 수 없습니다 :83
- ❗ 종속성 'egovframework.rte:egovframework.rte.fdl.idgnr:3.10.0'을(를) 찾을 수 없습니다 :88
- ❗ 종속성 'egovframework.rte:egovframework.rte.fdl.property:3.10.0'을(를) 찾을 수 없습니다 :93
- ❗ 플러그인 'org.apache.maven.plugins:maven-surefire-plugin:'을(를) 찾을 수 없습니다 :236
- ❗ 플러그인 'org.apache.maven.plugins:maven-surefire-plugin:'을(를) 찾을 수 없습니다 :237
- ❗ 플러그인 'org.apache.maven.plugins:maven-project-info-reports-plugin:3.0.0'을(를) 찾을 수 없습니다 :268
- ❗ 플러그인 'org.apache.maven.plugins:maven-project-info-reports-plugin:3.0.0'을(를) 찾을 수 없습니다 :269
- ❗ 플러그인 'org.apache.maven.plugins:maven-project-info-reports-plugin:3.0.0'을(를) 찾을 수 없습니다 :270
- ❗ 플러그인 'org.codehaus.mojo:surefire-report-maven-plugin:'을(를) 찾을 수 없습니다 :294
- ❗ 플러그인 'org.apache.maven.plugins:maven-javadoc-plugin:'을(를) 찾을 수 없습니다 :306

## 2. 샘플게시판 이해

### ❑ Maven Repository 에러

- [https://www.egovframe.go.kr/home/faqinfo/faqinfoRead.do?pagerOffset=0&searchKey=&searchValue=&menuNo=68&faqId=FAQ\\_00000000000000751](https://www.egovframe.go.kr/home/faqinfo/faqinfoRead.do?pagerOffset=0&searchKey=&searchValue=&menuNo=68&faqId=FAQ_00000000000000751)

A

\* 센트럴 레포지토리 주소

<https://repo1.maven.org/maven2/>

\* 표준프레임워크 레포지토리 주소

<https://maven.egovframe.go.kr/maven/>

다음과 같이 설정을 추가 합니다.

```
<repositories>
  <repository>
    <id>mvn2s</id>
    <url>https://repo1.maven.org/maven2/</url>
    <releases>
      <enabled>true</enabled>
    </releases>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </repository>
```

```
<repository>
  <id>egovframe</id>
  <url>https://maven.egovframe.go.kr/maven/</url>
  <releases>
    <enabled>true</enabled>
  </releases>
  <snapshots>
    <enabled>false</enabled>
  </snapshots>
</repository>
</repositories>
```



## 2. 샘플게시판 이해

### ❑ Maven Repository 에러

```
<repository>
  <id>egovframe</id>
  <url>http://maven.egovframe.go.kr/maven</url>
  <releases>
    <enabled>true</enabled>
  </releases>
  <snapshots>
    <enabled>>false</enabled>
  </snapshots>
</repository>
<repository>
  <id>egovframe_old1</id>
  <url>http://maven.egovframe.kr:8080/maven/</url>
  <releases>
    <enabled>true</enabled>
  </releases>
  <snapshots>
    <enabled>>false</enabled>
  </snapshots>
</repository>
<repository>
  <id>egovframe_old2</id>
  <url>http://www.egovframe.go.kr/maven/</url>
  <releases>
    <enabled>true</enabled>
  </releases>
  <snapshots>
    <enabled>>false</enabled>
  </snapshots>
</repository>
```



```
<repository>
  <id>egovframe</id>
  <url>https://maven.egovframe.go.kr/maven</url>
  <releases>
    <enabled>true</enabled>
  </releases>
  <snapshots>
    <enabled>>false</enabled>
  </snapshots>
</repository>
```

### 3. 표준프레임워크 버전 업그레이드

#### ❑ 실행환경 Migration 가이드 (3.10 -> 4.0)

- <https://www.egovframe.go.kr/wiki/doku.php?id=egovframework:rtemigration4.0>

#### ❑ pom.xml

- groupId: egovframework.rte → org.egovframe.rte
- artifactId: egovframework.rte.xxx → org.egovframe.rte.xxx
- version: 3.10.0 → 4.0.0

#### ❑ Maven Reposieoty

- <https://maven.egovframe.go.kr/maven/>

#### ❑ EgovBindingInitializer 관련 변경

- ```
public void initBinder(WebDataBinder binder) {
```

### 3. 표준프레임워크 버전 업그레이드

#### ❑ 패키지명 변경(egovframework.rte → org.egovframe.rte)

- resoreces/egovframework/spring/\*.xml
- src/main/webapp/WEB-INF/config/egovframework/\*.xml
- egovframework/sqlmap/example/\*.xml
- src/main/webapp/WEB-INF/web.xml

#### ❑ egovframework/spring/context-common.xml

```
<list>
  <value>classpath:/egovframework/message/message-common</value>
  <value>classpath:/org/egovframe/rte/fdl/idgnr/messages/idgnr</value>
  <value>classpath:/org/egovframe/rte/fdl/property/messages/properties</value>
</list>
```

#### ❑ 스프링 프레임워크 <form:form> 태그 변경

- src/main/webapp/WEB-INF/jsp/egovframework/example
- <form:form commandName="xxx"> → <form:form modelAttribute="xxx">

## 4. Spring Boot 전환

### ❑ pom.xml

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.4.5</version>
</parent>
```

```
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>3.1.0</version>
  <scope>provided</scope>
</dependency>
```

```
<plugin>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
```

## 4. Spring Boot 전환

### ❑ 파일 이동

- src/main/webapp/WEB-INF/config/egovframework/springmvc/dispatcher-servlet.xml  
→ src/main/resources/egovframework/springmvc/dispatcher-servlet.xml
- src/main/webapp/WEB-INF/config/egovframework/validator/validator.xml, validator-rules.xml  
→ src/main/resources/egovframework/validator/validator.xml, validator-rules.xml
- context-validator 수정(egovframework/spring/context-validator.xml)

```
<bean id="validatorFactory" class="org.springframework.validation.commons.DefaultValidatorFactory">
  <property name="validationConfigLocations">
    <list>
      <value>classpath:/egovframework/validator/validator-rules.xml</value>
      <value>classpath:/egovframework/validator/validator.xml</value>
    </list>
  </property>
</bean>
```

## 4. Spring Boot 전환

### ❑ 파일 생성

- /src/main/resources/application.properties

```
server.servlet.context-path=/boot  
spring.main.allow-bean-definition-overriding=true
```

- /src/main/java/egovframework/example/SampleBootApplication.java

```
package egovframework.example;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.boot.web.support.SpringBootServletInitializer;  
import org.springframework.context.annotation.ImportResource;  
  
@SpringBootApplication(scanBasePackageClasses = SampleBootApplication.class)  
@ImportResource({"classpath:/egovframework/spring/context-*.xml"  
    , "classpath:/egovframework/springmvc/dispatcher-servlet.xml"})  
public class SampleBootApplication extends SpringBootServletInitializer {  
    public static void main(String[] args) throws Exception {  
        SpringApplication.run(SampleBootApplication.class, args);  
    }  
}
```

### ❑ 호출

- <http://localhost:8080/boot/egovSampleList.do>

## 5. Gradle 전환

### ❑ build.gradle

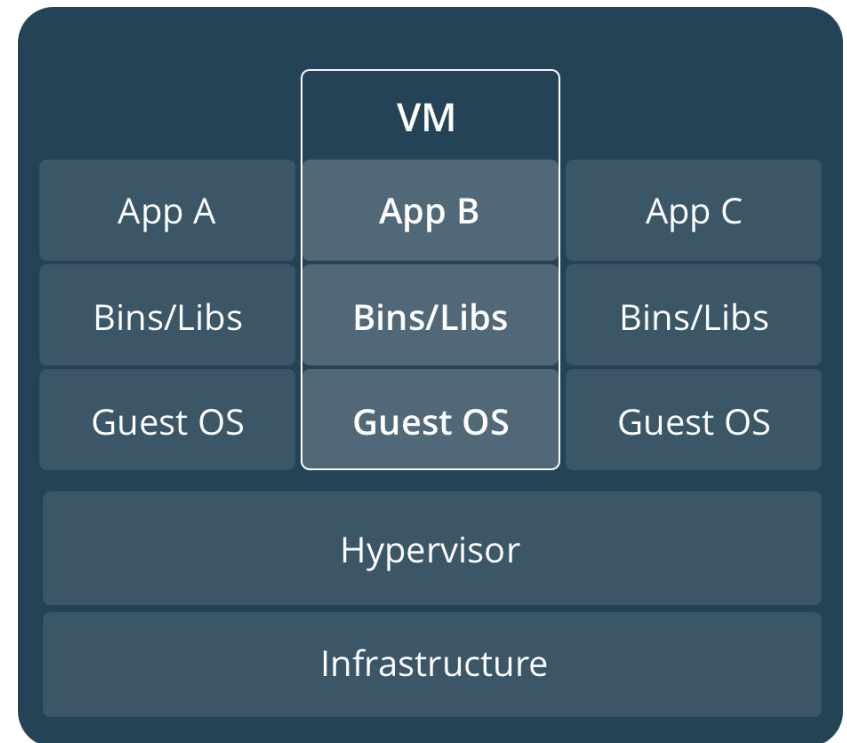
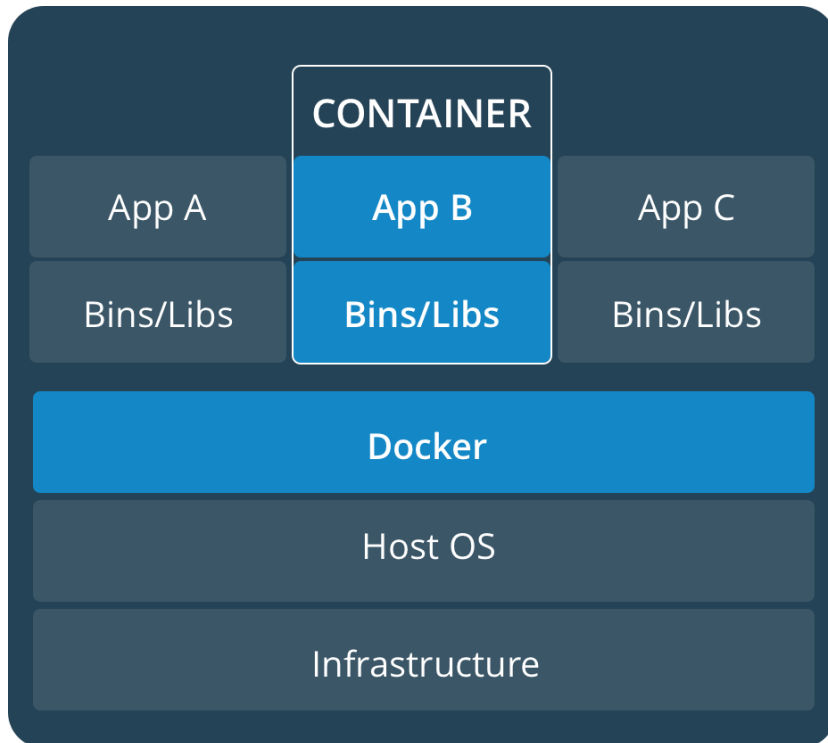
```
plugins {  
    id 'java'  
    id 'war'  
    id 'org.springframework.boot' version '2.4.5'  
    id 'io.spring.dependency-management' version '1.1.0'  
}  
  
sourceCompatibility = JavaVersion.VERSION_11  
targetCompatibility = JavaVersion.VERSION_11  
  
repositories {  
    maven { url 'https://repo1.maven.org/maven2/' }  
    maven { url 'https://maven.egovframe.kr/maven/' }  
    jcenter()  
}  
  
configurations {  
    all {  
        exclude group: 'ch.qos.logback', module: 'logback-classic'  
        exclude group: 'org.springframework.boot', module: 'spring-boot-starter-logging'  
    }  
}
```

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    implementation 'org.springframework.boot:spring-boot-starter-log4j2'  
    implementation 'org.springframework.boot:spring-boot-starter-jdbc'  
    implementation 'org.apache.tomcat.embed:tomcat-embed-jasper'  
  
    implementation ('org.egovframe.rte:org.egovframe.rte.ptl.mvc:4.0.0') {  
        exclude(group: 'org.egovframe.rte', module: 'spring-modules-validation')  
    }  
    implementation 'org.springframework.modules:spring-modules-validation:0.9'  
    implementation 'org.egovframe.rte:org.egovframe.rte.psl.dataaccess:4.0.0'  
    implementation 'org.egovframe.rte:org.egovframe.rte.fdl.idgnr:4.0.0'  
    implementation 'org.egovframe.rte:org.egovframe.rte.fdl.property:4.0.0'  
    compileOnly 'javax.servlet:javax.servlet-api:3.1.0'  
    implementation 'javax.servlet:jstl'  
    implementation 'taglibs:standard:1.1.2'  
    implementation 'org.antlr:antlr:3.5'  
    implementation 'org.hsqldb:hsqldb'  
}
```

## 6. Kubernetes 배포

### ❑ Docker 정의

- 개발자와 관리자가 컨테이너를 사용하여 애플리케이션을 개발, 배포 및 실행하기 위한 플랫폼
- 리눅스 컨테이너를 사용하여 응용 프로그램을 배포하는 것을 컨테이너화라고 함
- 컨테이너는 새로운 것은 아니지만, 애플리케이션을 쉽게 배치하기 위해 사용
- Containers and Virtual Machines



이미지 출처: <https://docs.docker.com/get-started/>



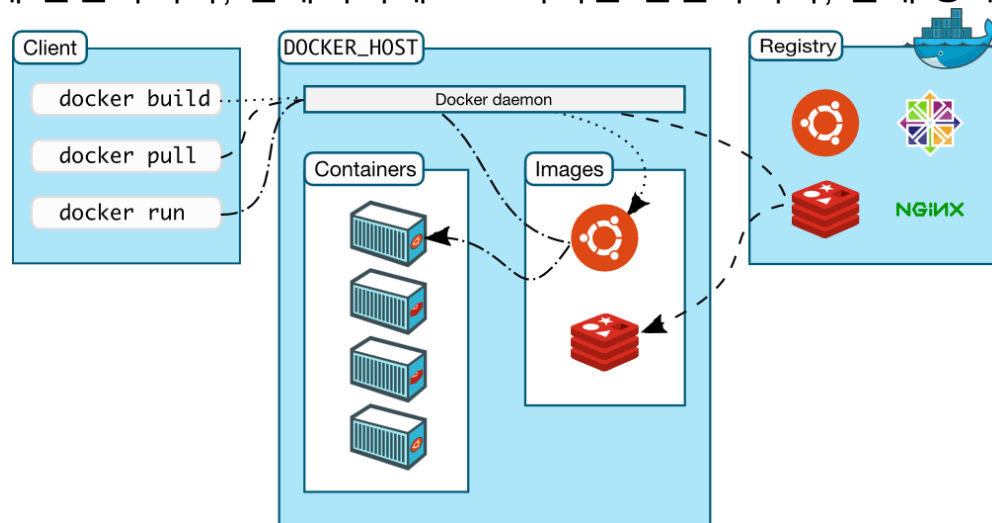
## 6. Kubernetes 배포

### ❑ IMAGES

- 도킹된 컨테이너를 생성하기 위한 지침이 포함된 읽기 전용 템플릿
- 종종 추가적인 사용자 지정과 함께 다른 이미지를 기반으로 함
- 다른 가상화 기술에 비해 이미지를 매우 가볍고 작고 빠르게 만드는 요소 중 하나임

### ❑ CONTAINERS

- 실행 가능한 이미지의 인스턴스
- DockerAPI또는 CLI를 사용하여 컨테이너를 생성, 시작, 중지, 이동 또는 삭제
- 하나 이상의 네트워크에 연결하거나, 컨테이너에 스토리지를 연결하거나, 현재 상태에 따라 새 이미지를 만들 수도 있음



이미지 출처: <https://docs.docker.com/engine/docker-overview/>

## 6. Kubernetes 배포

### ❑ Dockerfile 작성

- 이미지를 생성하기 위한 스크립트

```
FROM openjdk:11.0.16-jre-slim
LABEL maintainer=<dasomelli@gmail.com>
RUN ln -snf /usr/share/zoneinfo/Asia/Seoul /etc/localtime && echo Asia/Seoul > /etc/timezone

COPY ./build/libs/sample-3_10_0.war app.war
ENTRYPOINT ["java","-server","-D64","-Djava.security.egd=file:./dev/urandom","-jar","/app.war"]
EXPOSE 8080
```

Docker를 구동하기 위한 base image

Time Zone 설정

파일 복사

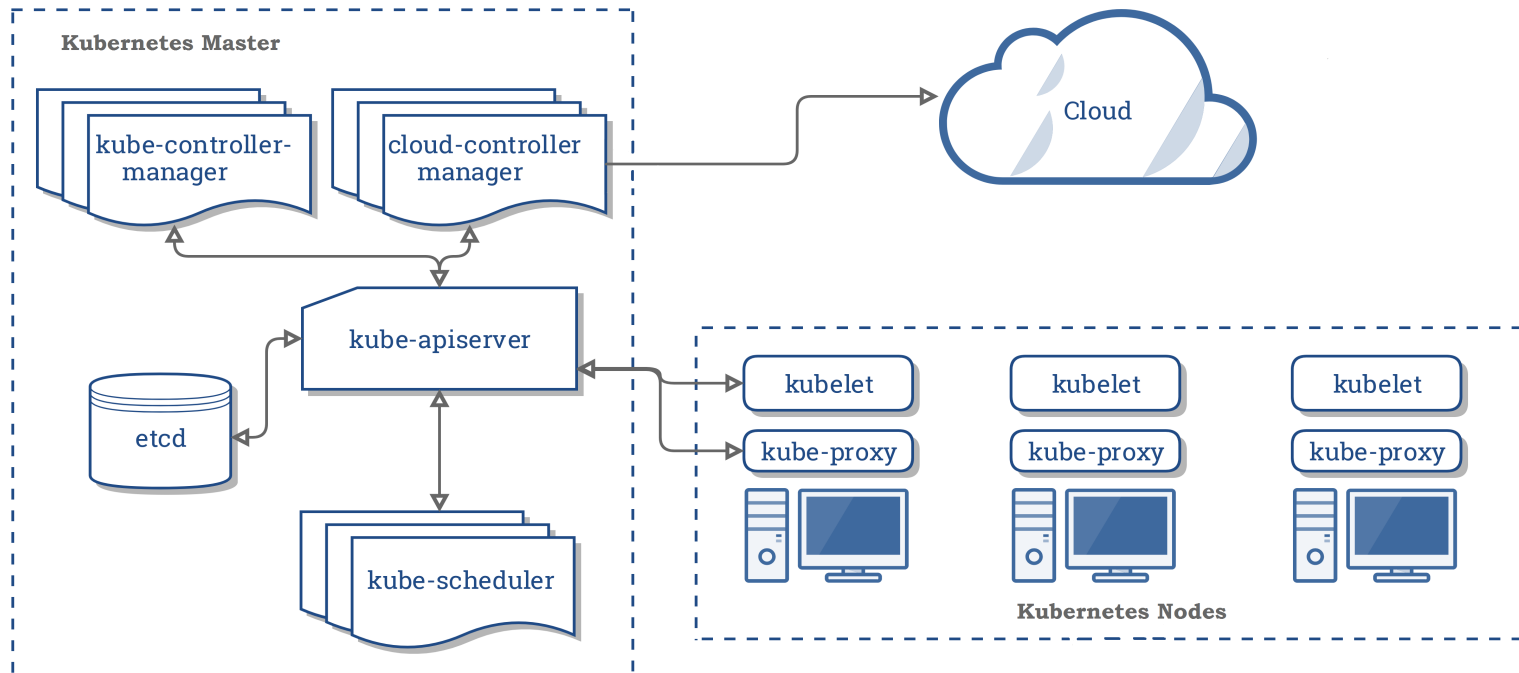
해당 포트 노출

컨테이너가 시작되었을 때 스크립트  
혹은 명령을 실행

## 6. Kubernetes 배포

### ❑ Kubernets 정의

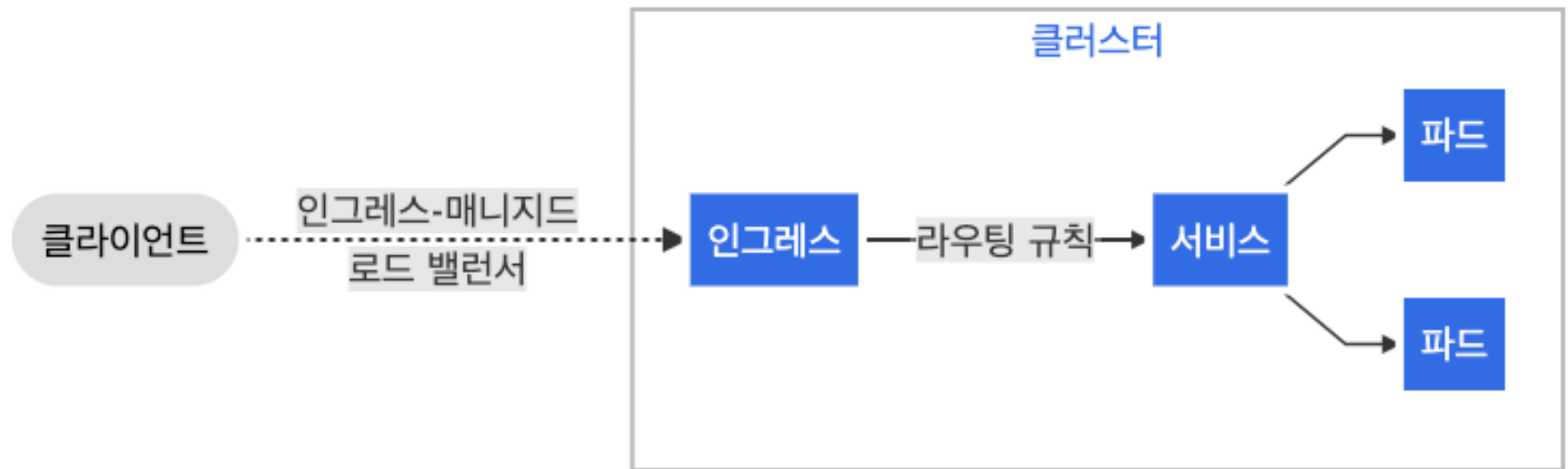
- 쿠버네티스(K8s)는 디플로이 자동화, 스케일링, 컨테이너화된 애플리케이션의 관리를 위한 오픈 소스 시스템으로서 원래 구글에 의해 설계되었고 현재 리눅스 재단에 의해 관리되고 있다.
- 목적은 여러 클러스터의 호스트 간에 애플리케이션 컨테이너의 배치, 스케일링, 운영을 자동화하기 위한 플랫폼을 제공하기 위함이다.
- 도커를 포함하여 일련의 컨테이너 도구들과 함께 동작한다. (위키백과)



## 6. Kubernetes 배포

### ❑ Ingress

- 인그레스는 클러스터 외부에서 클러스터 내부 서비스로 HTTP와 HTTPS 경로를 노출한다. 트래픽 라우팅은 인그레스 리소스에 정의된 규칙에 의해 컨트롤된다.
  - 인그레스는 외부에서 서비스로 접속이 가능한 URL, 로드 밸런스 트래픽, SSL / TLS 종료 그리고 이름 기반의 가상 호스팅을 제공하도록 구성할 수 있다.
- 인그레스 컨트롤러는 일반적으로 로드 밸런서를 사용해서 인그레스를 수행할 책임이 있으며, 트래픽을 처리하는데 도움이 되도록 에지 라우터 또는 추가 프론트 엔드를 구성할 수도 있다.



## 6. Kubernetes 배포

### ❑ Ingress

Ingress 오브젝트 명시

Ingress 이름

이름 기반의 가상 호스팅  
이름 기반의 가상 호스트는  
동일한 IP 주소에서 여러 호스트  
이름으로 HTTP 트래픽을  
라우팅하는 것을 지원한다.

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress
spec:
  ingressClassName: nginx
  rules:
    - host: opdc.io
      http:
        paths:
          - pathType: ImplementationSpecific
            path: /boot
            backend:
              service:
                name: sample
                port:
                  number: 8080
```

이 오브젝트를 생성하기 위해  
사용하고 있는 쿠버네티스 API

## 6. Kubernetes 배포

### ❑ Service

- Kubernetes의 서비스는 포드와 유사한 REST 객체입니다.

모든 REST 오브젝트와 마찬가지로 서비스 정의를 API 서버에 POST하여 새 인스턴스를 작성할 수 있습니다.

```
apiVersion: v1
kind: Service
metadata:
  name: sample
spec:
  selector:
    app: sample
  ports:
    - port: 8080
      targetPort: 8080
```

Service에서 사용하는 포트

Service에 연결될 Pod가 사용하는 포트

## 6. Kubernetes 배포

### ❑ Deployment

- 디플로이먼트 는 파드와 레플리카셋에 대한 선언적 업데이트를 제공한다.
- 디플로이먼트에서 의도하는 상태를 설명하고, 디플로이먼트 컨트롤러(Controller) 는 현재 상태에서 의도하는 상태로 비율을 조정하며 변경한다.  
새 레플리카셋을 생성하는 디플로이먼트를 정의하거나 기존 디플로이먼트를 제거하고, 모든 리소스를 새 디플로이먼트에 적용할 수 있다.

Pod 복제 개수(레플리카 파드)

배포 이미지(Docker)

IfNotPresent 또는 Never: 로컬 이미지 Always가 기본값

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: sample
  labels:
    app: sample
spec:
  replicas: 1
  selector:
    matchLabels:
      app: sample
  template:
    metadata:
      labels:
        app: sample
    spec:
      containers:
        - name: sample
          image: sample:0.0.2
          ports:
            - containerPort: 8080
          imagePullPolicy: IfNotPresent
      resources:
        requests:
          cpu: "0.5"
          memory: 0.5Gi
        limits:
          cpu: "0.5"
          memory: 0.5Gi
```

## 7. 참고자료

### ❑ Docker 설치

- <https://www.docker.com/products/docker-desktop>
- <https://docs.docker.com/docker-for-windows/install/>
- <https://docs.docker.com/docker-for-mac/install/>
- 정상 설치 확인
  - `docker version`

### ❑ Windows 10에 Linux용 Windows 하위 시스템 설치 가이드

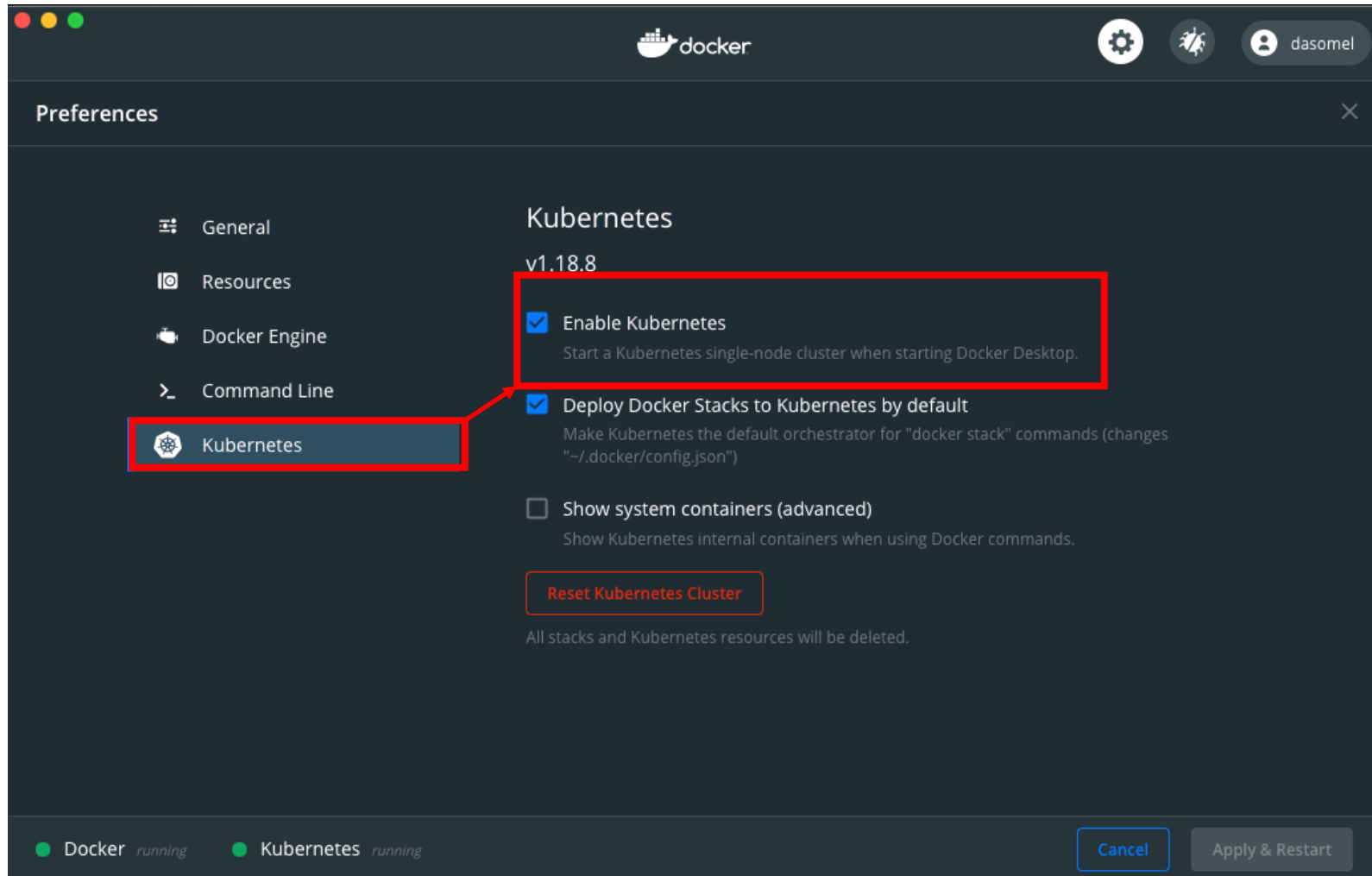
- WSL2(Windows Subsystem for Linux 2)
- 윈도우 사용자 필요시 참조
- <https://docs.microsoft.com/ko-kr/windows/wsl/install-win10>



## 7. 참고자료

### ❑ Kubernetes 설치

- docker-desktop 설치 후 Preferences → Kubernetes → Enable Kubernetes → Apply & Restart



## 7. 참고자료

### ❑ Infrastructure as Code

- 코드로서의 인프라스트럭처는 물리적 하드웨어 구성이나 인터페이스 구성 도구가 아닌 기계가 읽을 수 있는 정의 파일들을 통한 컴퓨터 데이터 센터의 관리 및 프로비저닝 과정이다(출처: 위키백과)

### ❑ Helm

- 정의: Kubernetes를 위한 패키지 매니저
- 설치 참고: <https://helm.sh/ko/docs/intro/install/>

### ❑ Ingress Controller 설치

- `helm repo add bitnami https://charts.bitnami.com/bitnami`
- `helm install ingress bitnami/nginx-ingress-controller`

## 7. 참고자료

### ❑ k9s

- Kubernetes 클러스터와 상호작용하는 터미널 UI를 제공  
응용 프로그램을 보다 쉽게 탐색, 관찰 및 관리  
지속적으로 변화를 감시, Resource와 상호작용하는 명령 제공  
(<https://github.com/derailed/k9s>)

```
Context: docker-desktop
Cluster: docker-desktop
User: docker-desktop
K9s Rev: 0.21.7 [8323]
K8s Rev: v1.18.8

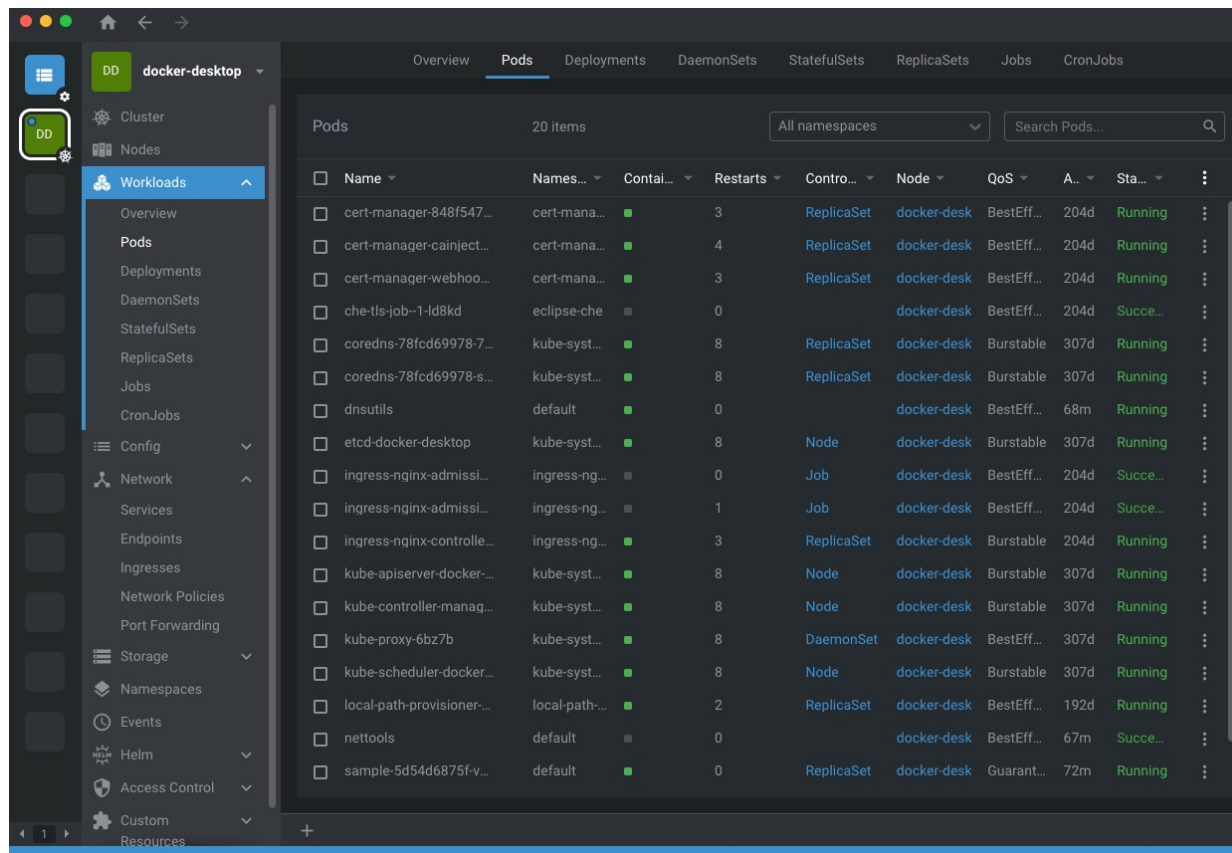
<0> all      <a> Attach    <l> Logs
<1> default  <ctrl-d> Delete  <shift-l> Logs Previous
<d> Describe <shift-f> Port-Forward
<e> Edit     <s> Shell
<?> Help    <f> Show PortFow
<ctrl-k> Kill <y> YAML

-----a-i-d-----
NAME                                PF  READY  RESTARTS  STATUS  IP          NODE          AGE
ingress-ingress-nginx-controller-855bd8cb4c-l9b77  ●  1/1      2 Running  10.1.0.69  docker-desktop  5d11h
keycloak-0                                  ●  1/1      0 Running  10.1.0.75  docker-desktop  7d14h
keycloak-1                                  ●  0/0      0 Pending  n/a        n/a            7h58m
mariadb-0                                  ●  1/1      2 Running  10.1.0.73  docker-desktop  13d
test-a-87d8c4c66-tbfdh                    ●  1/1      2 Running  10.1.0.70  docker-desktop  7d11h
test-a1-7df75bcd9c-ghfqm                   ●  1/1      2 Running  10.1.0.71  docker-desktop  7d10h
test-c-8558b9c6c7-ng5tp                    ●  1/1      2 Running  10.1.0.72  docker-desktop  4d13h
test-c1-7f965b589-4nhs7                    ●  1/1      2 Running  10.1.0.74  docker-desktop  4d13h
```

## 7. 참고자료

### □ openlens

- The Kubernetes IDE
- 기존 LENS가 유료화됨에 따라 Lens Open Source Project  
(<https://github.com/MuhammedKalkan/OpenLens>)



□ **91차 세미나(2020.12.01)-나도 해보자! Hands-on Cloud on eGov-Framework**

- <https://open.egovframe.org/oc/products/seminarItem.do?nttId=19932&pageIndex=2>