

Operator – SDK

애플리케이션을 패키징, 배포 및 관리하는 것

컨트롤러, 커스텀 리소스 생성, 등록을 자동화해 놓은 것

1) CRD(Custom Resource Definition) 란?

직접 리소스의 종류를 정의해서 사용할 수 있음.

2) K8s 컨트롤러

선언형(Declarative) 방식 : 최종적으로 도달해야 하는 **바람직한 상태(Desired State)**를 정의 한 뒤, **현재 상태(Current State)**가 바람직한 상태와 다를 경우 이를 일치하도록 만드는 방법이다.

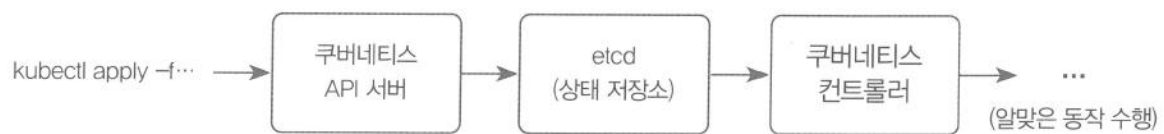
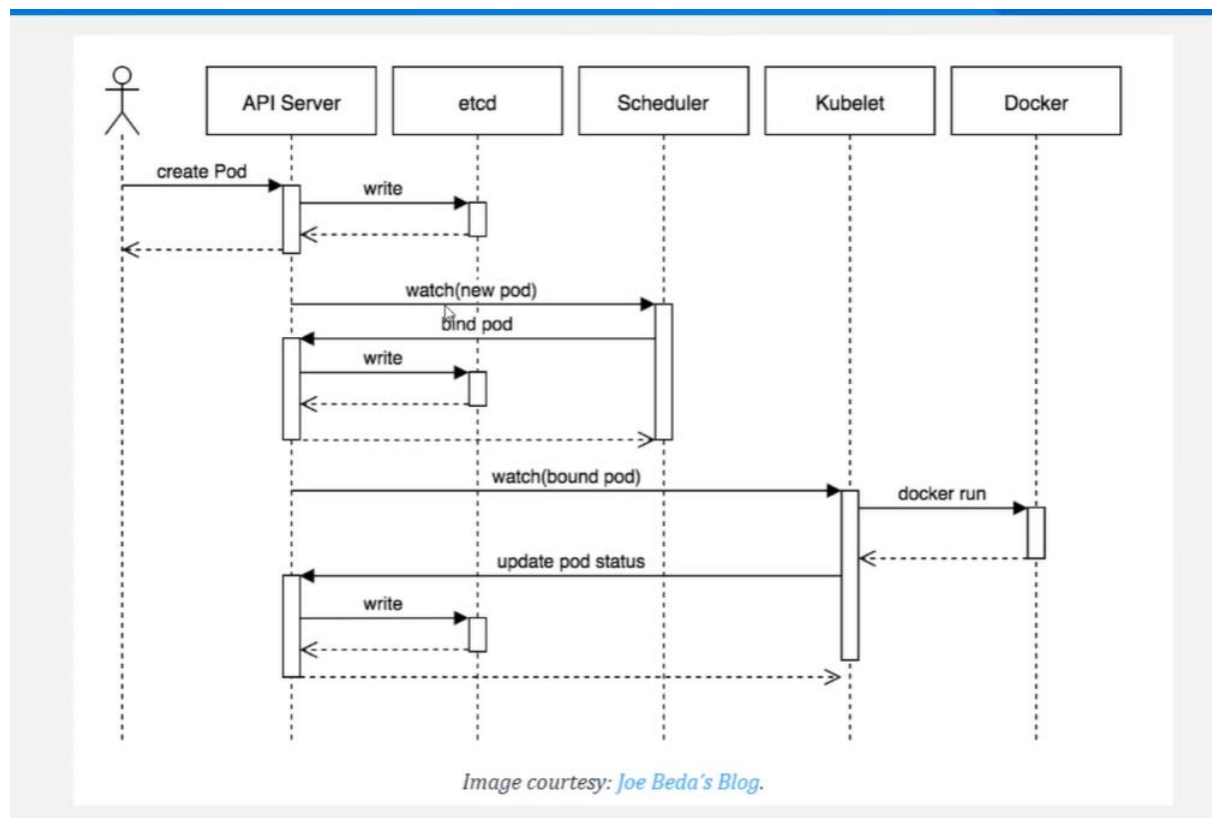
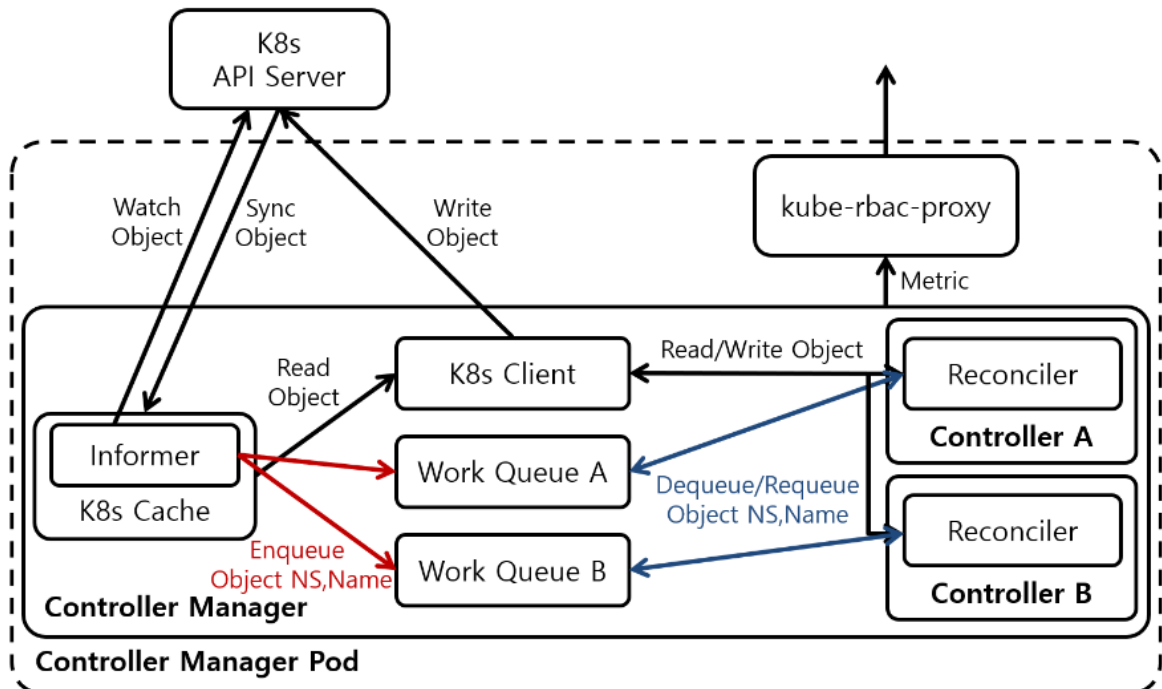
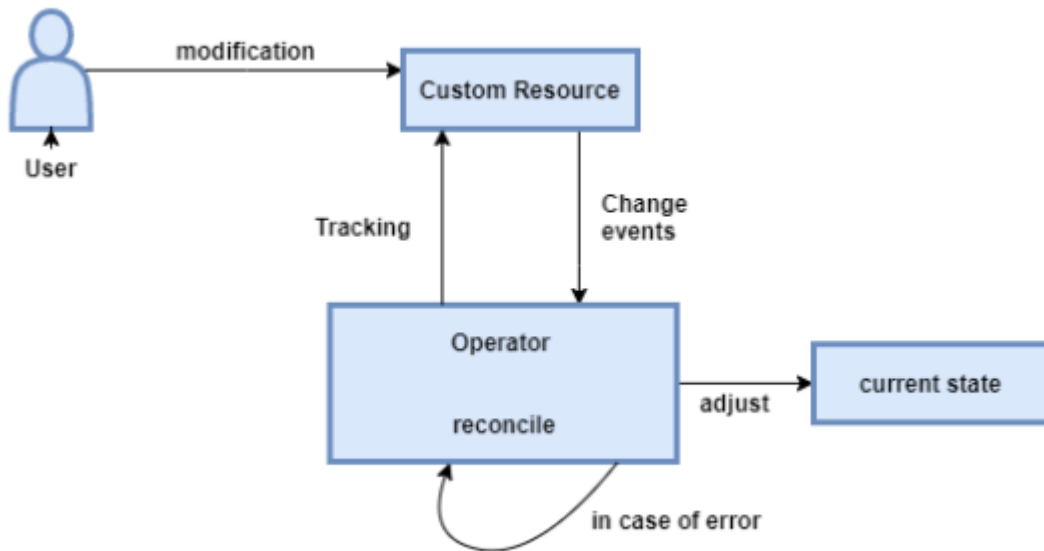


그림 12.2 선언형 방식의 예

- 3) Operator 패턴 : CRD를 사용할 수 있도록 컨트롤러를 구현하는 방법
- 4) Reconcile: 현재 상태가 desired State 가 되도록 특정 동작을 수행 하는 것



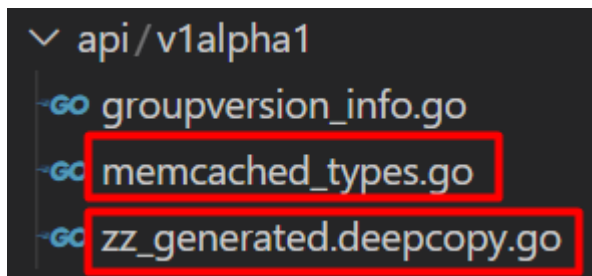
[그림 1] Controller Manager Architecture

1. Operator-sdk 구조

1) Crd spec 정의 (api -> crd)

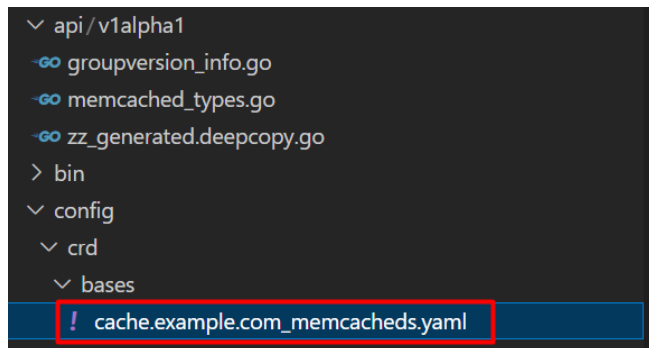
```
$ make generate
```

Memcached_type 파일은 기본적인 내용만 있어서 상세 내용 추가하여
zz_generated.deepcopy.go 파일을 생성한다.



```
$ make manifests
```

zz_generated.deepcopy.go 기반으로 config base 파일이 만들어짐



Crd manifest 파일 생성

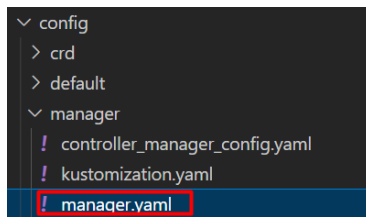
```
$ make install
```

Crd 가 api server 에 전달되서 etcd에 저장된다.

Config -> crd -> bases -> etcd 에 등록

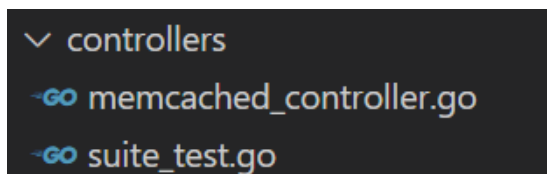
2) Controller 생성

Config 디렉토리에 manager.yaml 파일 내용대로 만들어진다.



```
apiVersion: v1
kind: Namespace
metadata:
  labels:
    control-plane: controller-manager
  name: system
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: controller-manager
  namespace: system
  labels:
    control-plane: controller-manager
spec:
  selector:
    matchLabels:
      control-plane: controller-manager
  replicas: 1
  template:
    metadata:
      annotations:
        kubectrl.kubernetes.io/default-container: manager
      labels:
        control-plane: controller-manager
```

Namespace와 deployment 가 선언되어 있다.



Controller.go 파일도 reconcile, createdeployment 함수 추가해서 수정한다.

3) Docker registry 생성

도커 이미지 이름 변경 : docker registry/imagename : tag