



상위 1% 월급쟁이를 위한



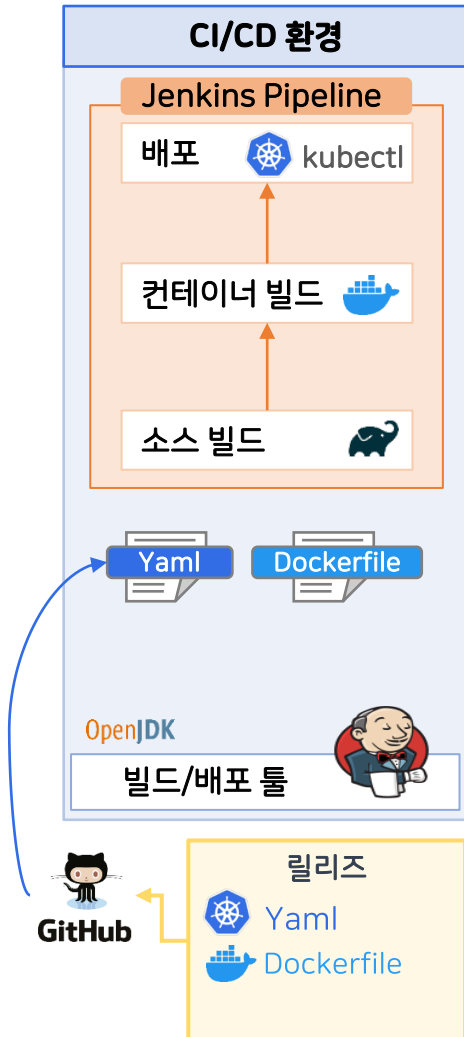
지상편

# 쿠버네티스 어나더 클래스

**Jenkins Pipeline**  
(기초부터 Blue/Green까지)

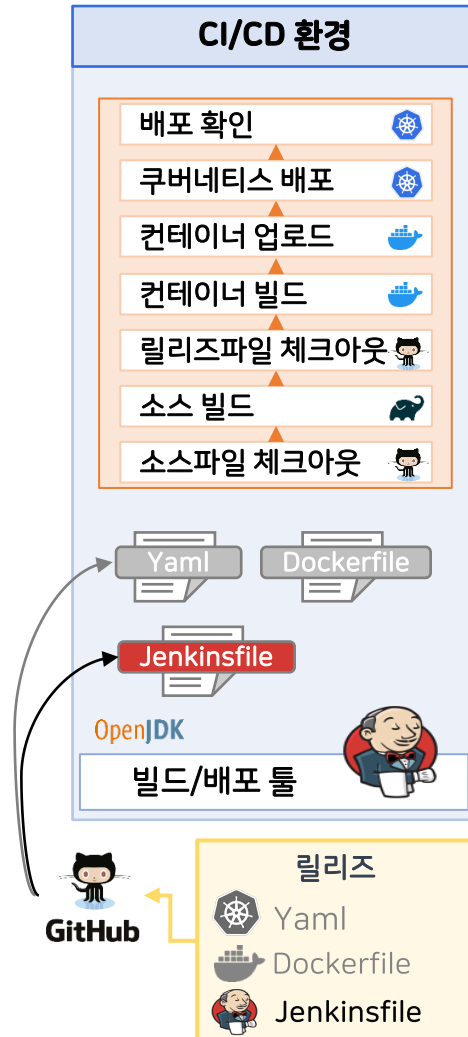
## Step1

Jenkins Pipeline  
기본 구성 만들기



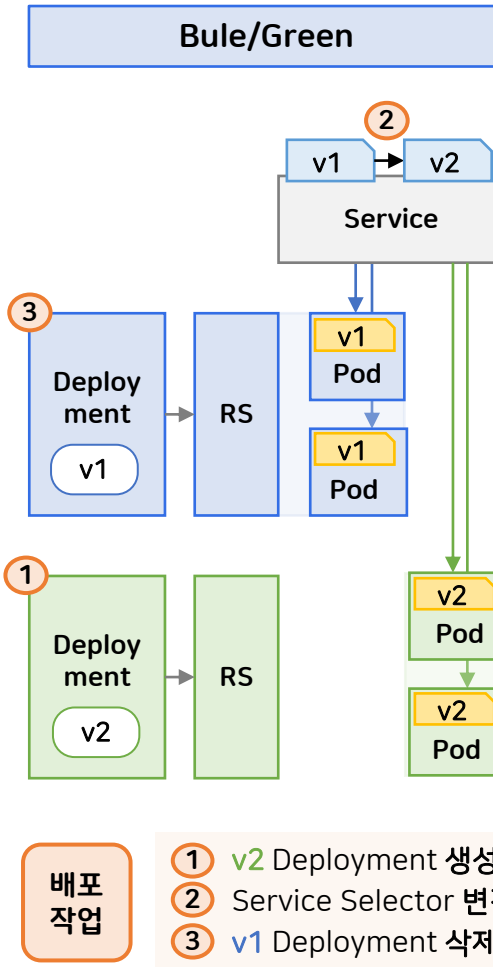
## Step 2

Github 연결 및 파이프라인 세분화



## Step 3

Blue/Green 배포 만들기 및  
특징 실습



## 특징

### 확인 3-2

- 수동 배포 시 롤백이 빠름

### Step 4

- Script를 통해 자동 배포 가능

### 실습 x

- v2에 과도한 트래픽 유입시 문제 발생

## 유즈 케이스

### 확인 3-1

운영에서만 테스트  
가능한 경우

QA 담당자



# 단계별로 구축해보는 배포 파이프라인

Jenkins Pipeline - 기초부터 Blue/Green 까지

```
pipeline {  
  agent any  
  
  stages {  
    stage('Source Build') {  
      steps {  
        // 실행 스크립트  
      }  
    }  
    stage('Container Build') {  
      steps {  
        // 실행 스크립트  
      }  
    }  
  }  
}
```

```
node('slave') {  
  stage('Source Build') {  
    steps {  
    }  
  }  
}
```

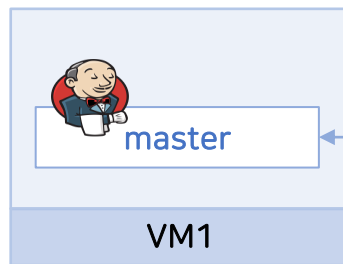
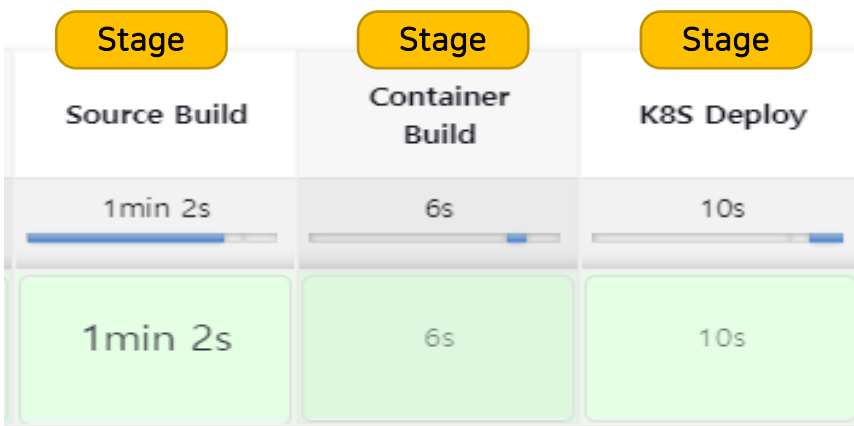
컨테이너 실행



Dockerfile 빌드



Yaml 배포

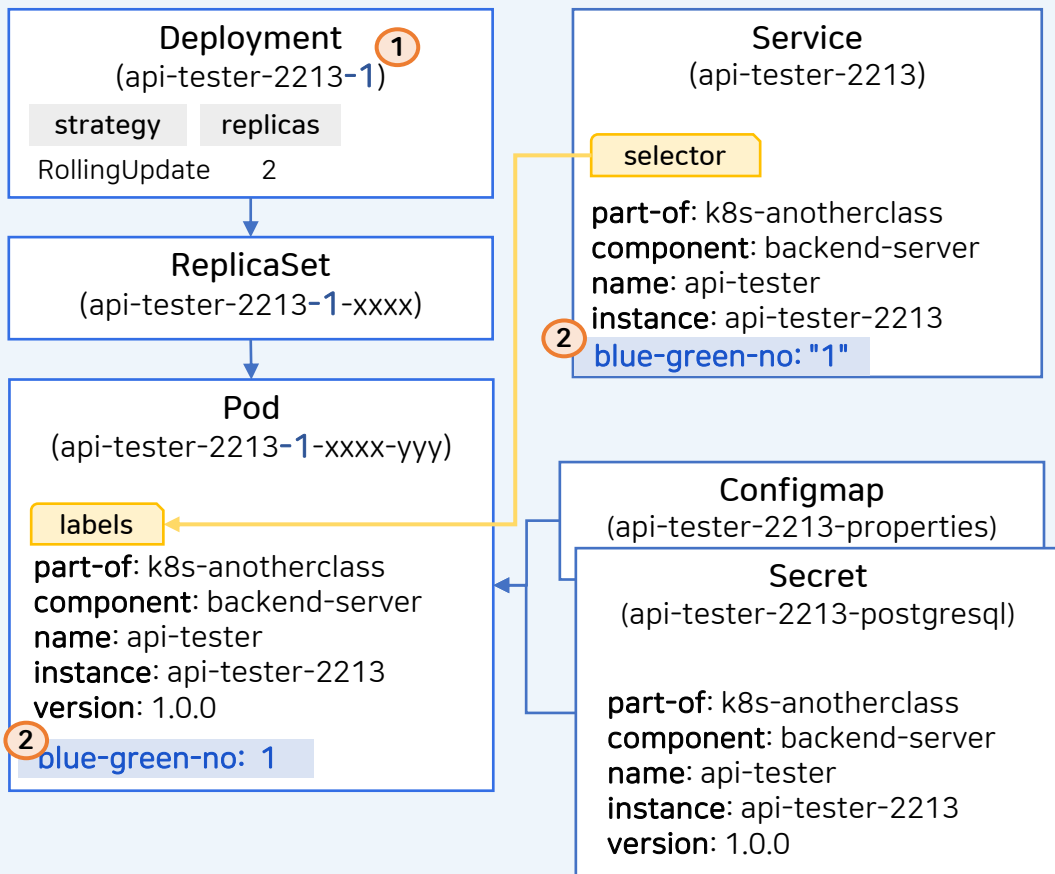




# Blue/Green시 고려해야 하는 요소

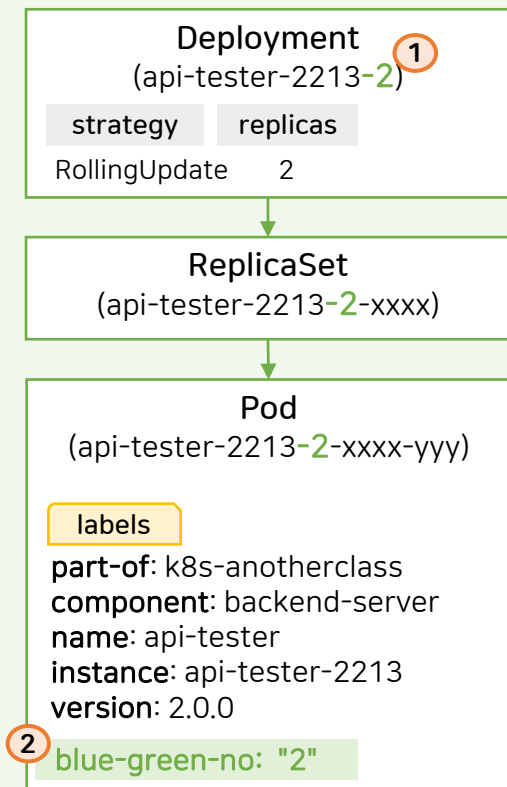
## Blue

Namespace (anotherclass-221)



## Green

Namespace (anotherclass-221)



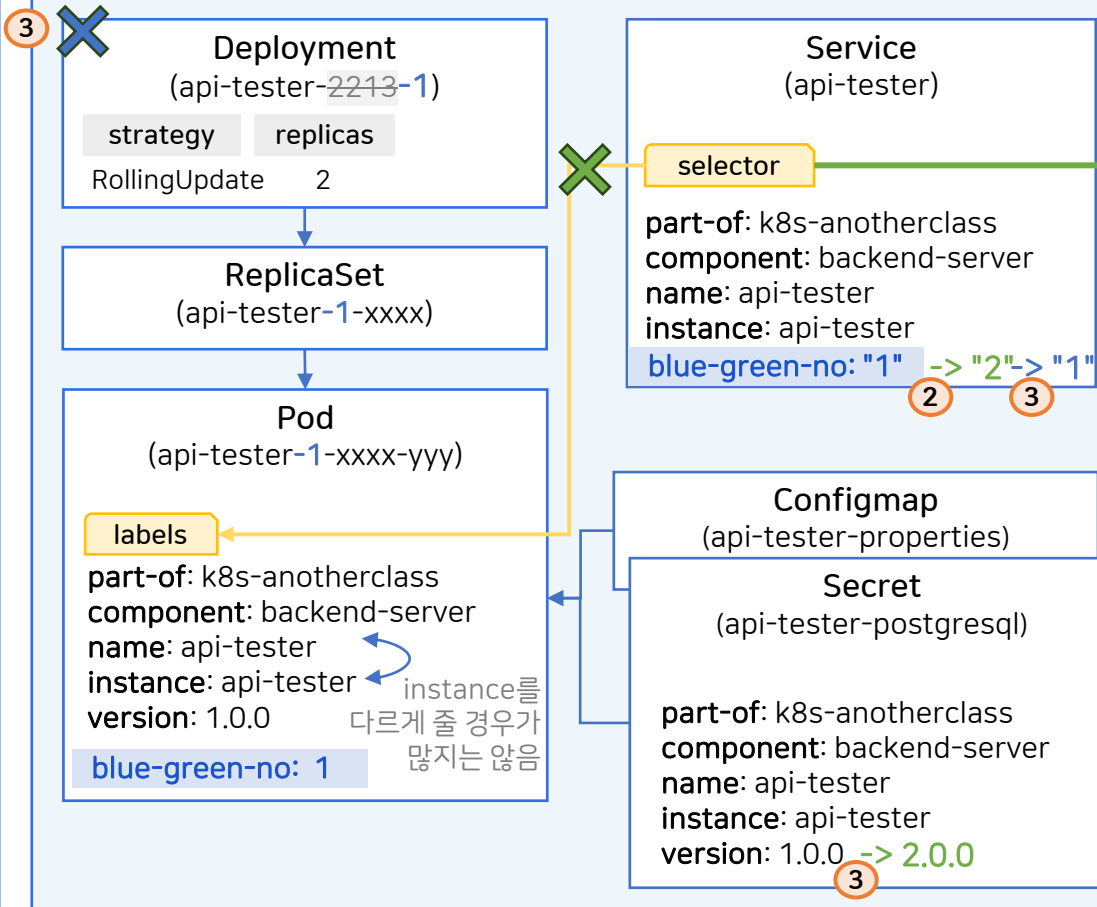
1. Blue-Green 배포를 고려한 **Deployment 네이밍** 필요
2. Blue-Green 배포를 위한 추가 **레이블 및 셀렉터** 필요



# Blue/Green 수동 배포 해보기

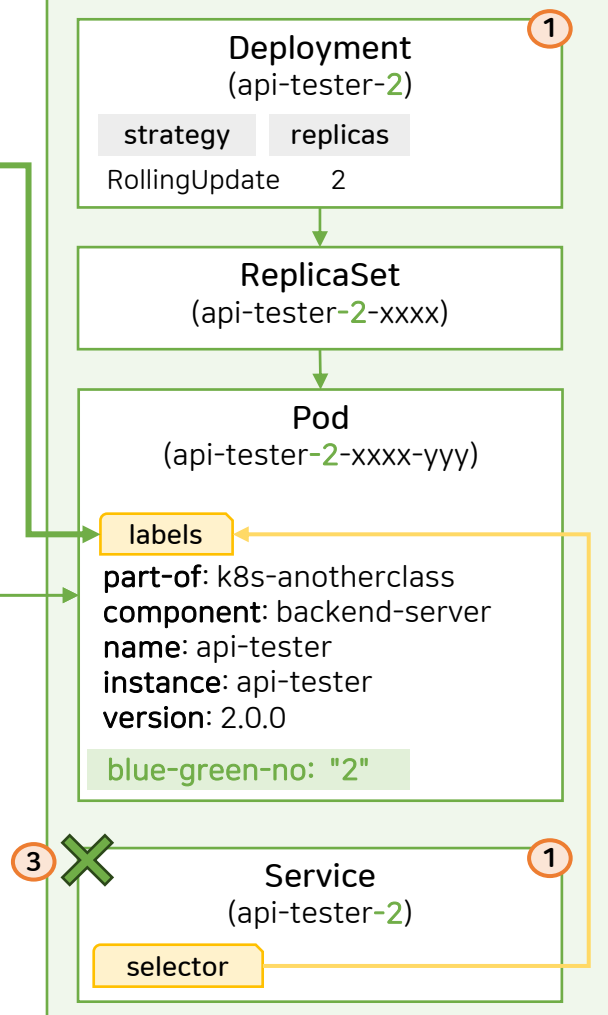
## Blue

Namespace (anotherclass-221)



## Green

Namespace (anotherclass-221)



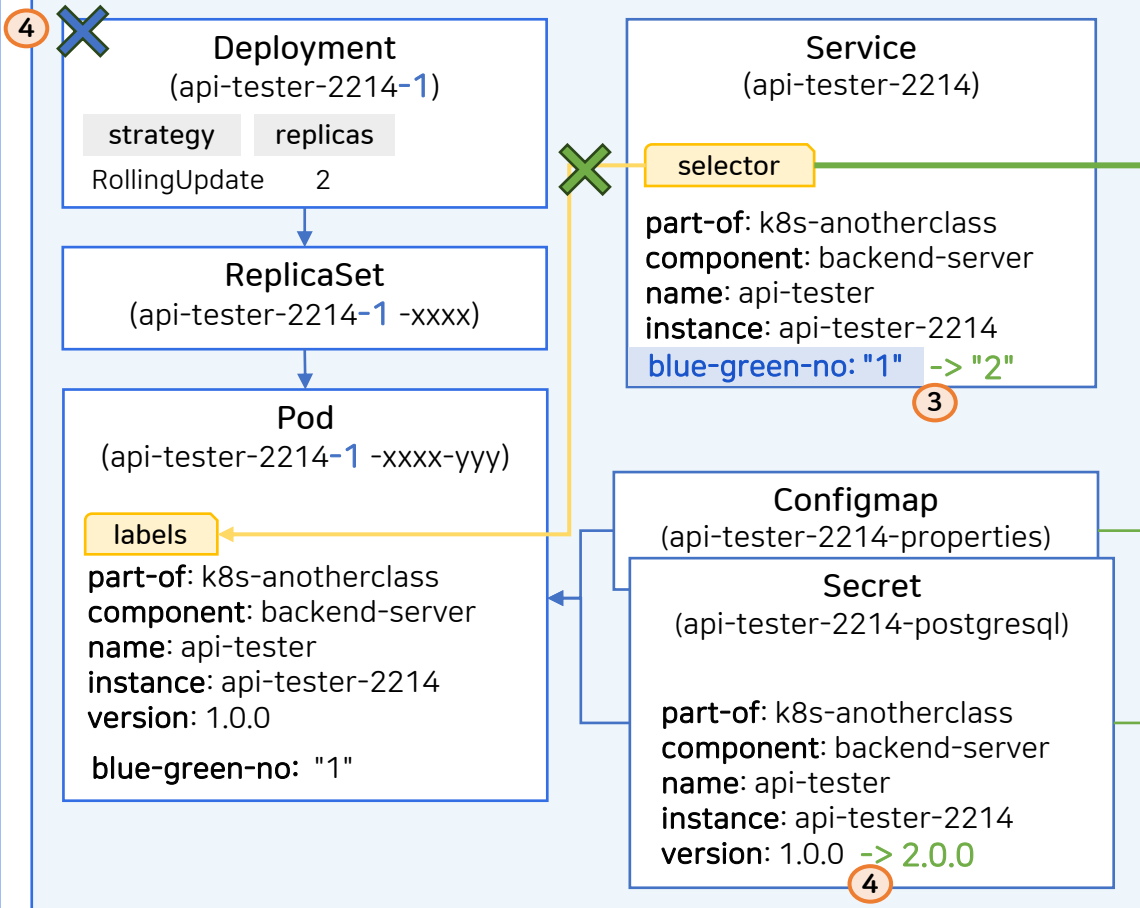
1. **Green Deployment** 생성  
테스트를 위한 **Service** 생성
2. Service의 Selector (**blue-green-no**)를 **2**로 변경하여 트래픽을 Green으로 전환
3. **Blue Deployment**, **Green Service** 삭제  
및 관련 모든 리소스의 **레이블 정보 변경** (version)  
or **Rollback**



# Blue/Green 자동배포 Script 만들기

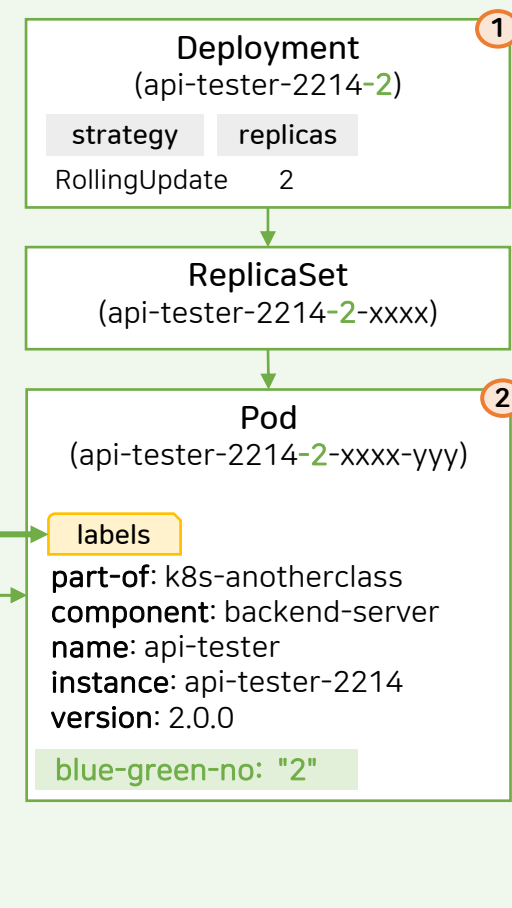
## Blue

Namespace (anotherclass-221)



## Green

Namespace (anotherclass-221)



1. Green Deployment 생성
2. Pod를 조회해서 Ready 상태인지 확인
3. Service의 Selector (blue-green-no)를 2로 변경하여 트래픽을 Green으로 전환
4. Blue Deployment 삭제 및 관련 모든 리소스의 레이블 정보 변경 (version)