

# 5 Kubernetes 배포를 위한 GitHub와 ArgoCD 활용

## 01 Git을 활용한 Kubernetes 배포 방법 소개

# Kubernetes 배포를 위한 GitHub와 ArgoCD 활용

**01.**  
Git을 활용한  
Kubernetes  
배포 방법 소개

1. Git을 활용한 Kubernetes 배포 방법 소개
2. [실습] Github Repository 생성 및 설정
3. Kubernetes 배포 도구(ArgoCD, ArgoRollout) 소개
4. [실습] ArgoCD 구성
5. [실습] ArgoCD를 활용한 Kubernetes Manifest 배포
6. [실습] ArgoRollout 구성
7. [실습] Argo Rollout을 활용한 무중단 배포

# Git을 활용한 Kubernetes 배포 방법의 아이디어

## Kubernetes 배포 방법

- K8s는 **Manifest** 기반의 **선언적** 코드를 명시해 **배포**할 대상을 특정 **Object**로 구현

## Git 활용

- 코드를 저장하고 **버전**을 관리하여 코드의 **형상**을 관리 할 수 있음

## Git을 활용한 Kubernetes 배포 방법의 아이디어

- **Git**의 저장한 Manifest 기반의 **선언적** 코드로 Kubernetes **관리**가 가능하지 않을까?
- Manifest 기반의 **선언적** 코드들은 Git에서 **버전** 관리가 가능하지 않을까?
- K8s Manifest로 배포된 K8s **Object**의 실행 상태의 **형상** 관리가 가능하지 않을까?

# GitOps란?

## 01. Git을 활용한 Kubernetes 배포 방법 소개

### GitOps의 기원

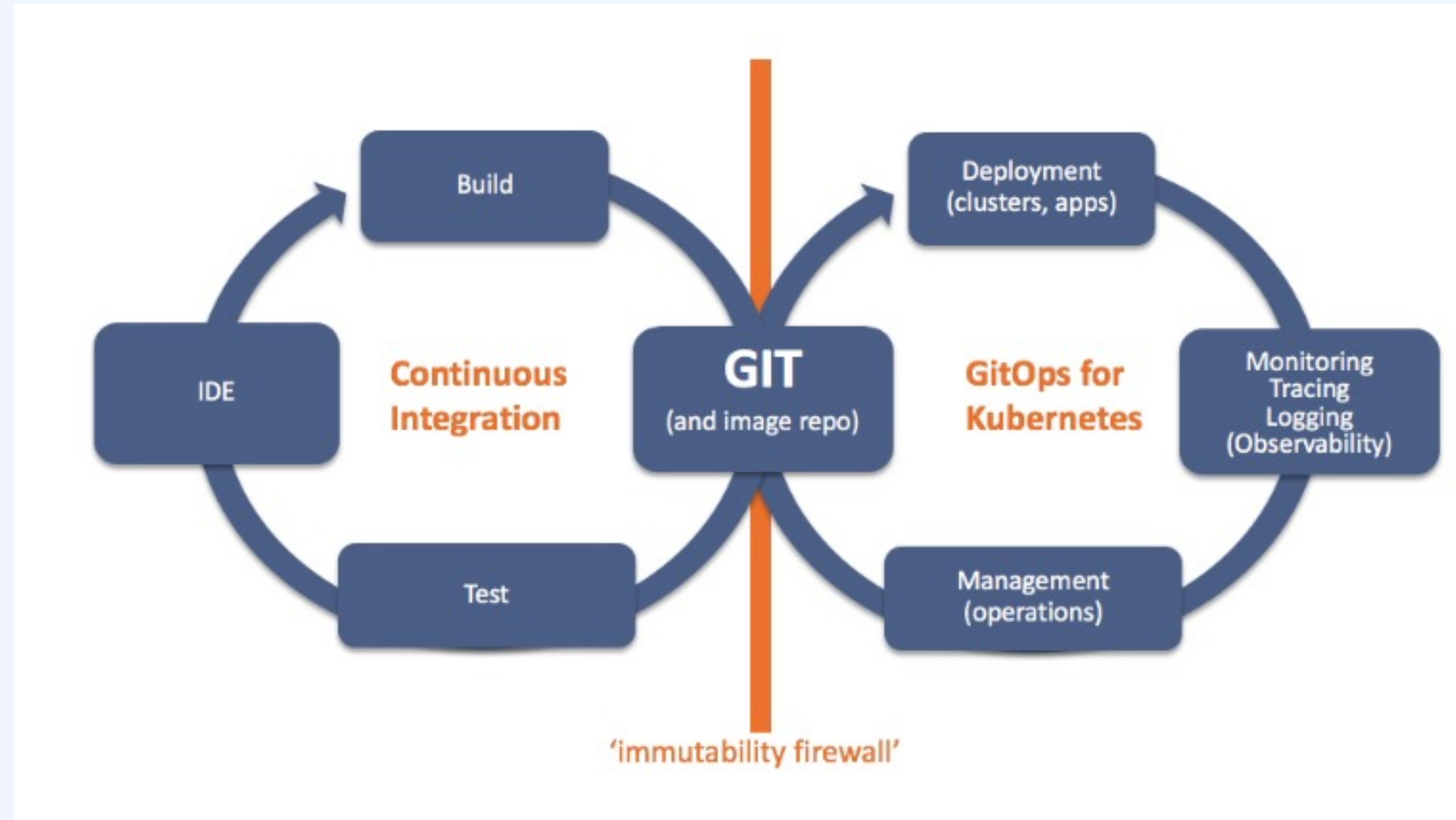
GitOps 개념은 2017년 **Weaveworks**에서 최초로 제안함

### GitOps의 목표

1. 시스템 외부의 모델을 **소스**로 사용하여 Kubernetes의 **운영**을 자동화
2. GitOps를 **활용**해 K8s 기반의 플랫폼, 서비스, 앱을 **관리**하는 데 사용
3. 시스템과 **상호 작용**하는 다양한 이해 관계자를 고려하고 업무를 명확하게 **분리**

# Git기반 CI/CD 및 운영 사이클

## 01. Git을 활용한 Kubernetes 배포 방법 소개



출처 : <https://www.weave.works/blog/what-is-gitops-really>

# GitOps 원칙 #1

01.  
Git을 활용한  
Kubernetes  
배포 방법 소개

- OpenGitOps 기준으로 GitOps를 구현할 때의 표준 및 구조화된 방식을 제안
  - <https://opengitops.dev/>
- 선언적 원하는 상태(Desired State)원칙
  - GitOps 관리시스템은 사용자와 플랫폼이 모두 읽고 쓸 수 있는 선언적 코드로 표현된 원하는 상태(Desired State)를 가져야 함
- 불변한 원하는 상태(Desired State)버전 원칙
  - 원하는 상태(Desired State)는 버전 관리 및 버전의 불변성을 지원하고 완전한 버전 기록을 유지하는 방식으로 저장해 형상을 관리

## GitOps 원칙 #2

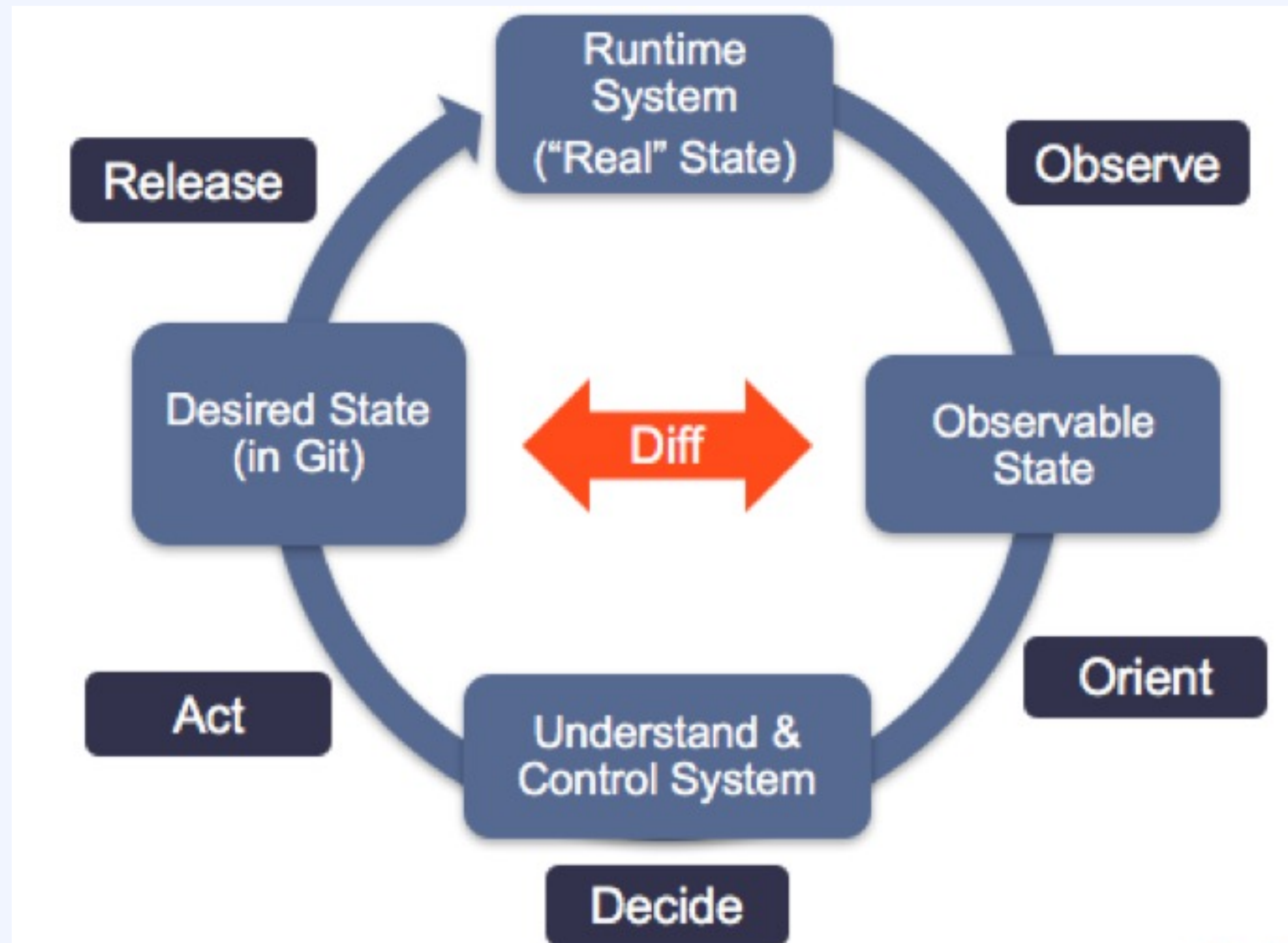
**01.**  
Git을 활용한  
Kubernetes  
배포 방법 소개

- **지속적인 상태 조정 원칙**
  - GitOps 관리 시스템은 플랫폼의 실제 **배포된 상태**를 원하는 상태(Desired State)와 지속적으로 **자동**으로 비교함
  - 실제 배포된 상태와 원하는 상태(Desired State)가 다른 경우, 이를 **일치**시키기 위한 **자동화** 작업 진행
- **선언을 통한 작동 원칙**
  - 플랫폼이 의도적으로 작동되는 유일한 **메커니즘**은 GitOps **원칙**을 통해서만 가능



# GitOps 상태 사이클

## 01. Git을 활용한 Kubernetes 배포 방법 소개



출처 : <https://www.weave.works/technologies/gitops/>



# GitOps 관리를 위한 Git Repository 구성

**01.**  
Git을 활용한  
Kubernetes  
배포 방법 소개

- **GitOps Repository**
  - 아래 3개의 Repository를 **제어, 실행**을 위한 **코드** 관리
- **Platform Repository**
  - 플랫폼 및 Kubernetes 프로비저닝을 위한 **IaC 코드** 관리
- **Management Repository**
  - Kubernetes 관리를 위한 설정, 플러그인, 시스템 **배포 코드** 관리
- **Service Repository**
  - 컨테이너 기반의 서비스 앱 개발 **소스코드** 관리

# GitOps 관리를 위한 Git Repository 구성도

**01.**  
Git을 활용한  
Kubernetes  
배포 방법 소개

